



## Calhoun: The NPS Institutional Archive DSpace Repository

---

Theses and Dissertations

1. Thesis and Dissertation Collection, all items

---

1973

### A queueing MODEL simulation with dynamic graphical display.

Riddell, Cyrus Michael.

Monterey, California. Naval Postgraduate School

---

<http://hdl.handle.net/10945/16769>

---

*Downloaded from NPS Archive: Calhoun*



<http://www.nps.edu/library>

Calhoun is the Naval Postgraduate School's public access digital repository for research materials and institutional publications created by the NPS community.

Calhoun is named for Professor of Mathematics Guy K. Calhoun, NPS's first appointed -- and published -- scholarly author.

Dudley Knox Library / Naval Postgraduate School  
411 Dyer Road / 1 University Circle  
Monterey, California USA 93943

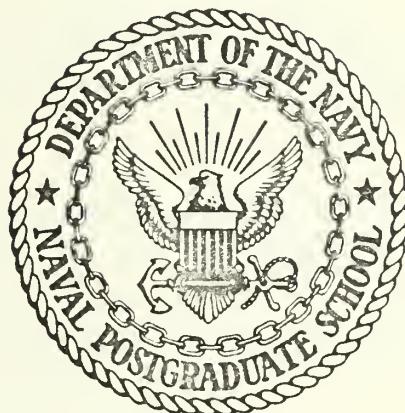
A QUEUING MODEL SIMULATION  
WITH DYNAMIC GRAPHICAL DISPLAY

Cyrus Michael Riddell

Library  
Naval Postgraduate School  
Monterey, California 93940

# NAVAL POSTGRADUATE SCHOOL

## Monterey, California



# THESIS

A QUEUING MODEL SIMULATION  
WITH DYNAMIC GRAPHICAL DISPLAY

by

Cyrus Michael Riddell

Thesis Advisor:

R. W. Butterworth

March 1973

*Approved for public release; distribution unlimited.*

T153548



A Queuing Model Simulation  
with Dynamic Graphical Display

by

Cyrus Michael Riddell  
Lieutenant, United States Navy  
B.A., University of Washington, 1965

Submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL  
March 1973



## ABSTRACT

The purpose of this thesis is to provide the student of queuing systems with a vehicle through which a better understanding of the interrelationships between stochastic processes and queuing systems can be achieved.



## TABLE OF CONTENTS

I.	INTRODUCTION	- - - - -	7
II.	QUEUING MODEL SIMULATION	- - - - -	9
	A. THE QUEUING MODELS	- - - - -	10
	1. G/G/C-	- - - - -	10
	2. G/G/C with Losses-	- - - - -	11
	3. G/G/C with Finite Source	- - - - -	12
	4. G/G/C with Feedback to the Queue	- - - -	13
	5. Two Simultaneous G/G/C Queues-	- - - -	14
	B. ARRIVAL/SERVICE DISTRIBUTIONS AND QUEUE DISCIPLINES-	- - - - -	14
	1. Distributions-	- - - - -	15
	a. K-ERLANG	- - - - -	15
	b. Hyperexponential	- - - - -	15
	c. Degenerate	- - - - -	16
	2. Queuing Disciplines-	- - - - -	16
	a. FIFO	- - - - -	16
	b. LIFO	- - - - -	16
	c. SSTF	- - - - -	16
	d. Random Draw-	- - - - -	17
	3. Length of the Simulation	- - - - -	17
III.	THE METHOD	- - - - -	18
	A. THE COMPUTER SIMULATION	- - - - -	18
	B. THE GRAPHICAL DISPLAY	- - - - -	18
IV.	STATISTICS FROM THE SIMULATION	- - - - -	20



V.   PARAMETER INPUT- - - - -	22
A.   SAMPLE INPUT CARD- - - - -	22
B.   NAMELIST INPUT VIA THE GRAPHICS TERMINAL - -	23
C.   INPUT VARIABLES- - - - -	23
VI.   THE PROGRAM- - - - -	25
APPENDIX A - ANALOG SCHEMATIC- - - - -	28
COMPUTER PROGRAM LISTING - - - - -	29
INITIAL DISTRIBUTION LIST- - - - -	87
FORM DD 1473 - - - - -	89



## LIST OF PICTURES

1. G/G/C Queuing System	- - - - -	10
2. G/G/C Queuing System with Losses	- - - - -	11
3. G/G/C Queuing System with Finite Source	- - - - -	12
4. G/G/C Queuing System with Feedback	- - - - -	13
5. Two Simultaneous G/G/C Queuing Systems	- - - - -	14
6. Statistics Page	- - - - -	20
7. Parameters Page	- - - - -	22



#### ACKNOWLEDGEMENTS

The Hybrid Computer at the Naval Postgraduate School is a self-operated system; because many of the complexities of the system were unfamiliar to the programmer, the technical assistance of Mr. R. Limes, Mr. W. Thomas, and Mr. A. Wong was invaluable.



## I. INTRODUCTION

A queuing system is comprised of action facilities (servers) and waiting facilities (queues or lines). The relationship between the components can be of many forms, each with their own characteristics. Consider a tool crib in a machine shop where the machinists are obliged to check out their tools; or consider the telephone system, in which callers request communication capacity. Both situations involve requests for action from limited resources, in these cases, servers or telephone circuits. In order to allocate resources properly, it is necessary to understand the interaction between requests for action and the actions themselves. Individually these can be modeled as stochastic processes, but in order to predict flow through the queuing system, it is necessary to understand the interrelation between the patterns of event occurrences. When demands occur at regular intervals and actions are of fixed duration, it is relatively easy to predict flow; if there are more requests per unit of time than actions over the same time span, the queue will grow without bound unless the event relationship is changed. When variation is added to the timing of events, the flow is not as obvious. For example, if the average number of requests for action equals the average number of actions, per unit of time, but both occur with variability, then it would be useful to be able to predict such quantities as expected delay and action



facility idle time. Queuing models are used for this purpose.

The purpose of this thesis is to provide the student of queuing systems with a vehicle through which a better understanding of the interrelationships between stochastic processes and queuing systems can be achieved.

This is done through a combination of a computer simulation and a computer-driven graphical display.



## II. QUEUING MODEL SIMULATION

This computer program simulates queuing systems in two stages. First, a next-event type computer simulation creates a time history of the queuing system. Second, a dynamic graphical display depicts the occurrence of the queuing process in real time.

Five basic, academically interesting, queuing models, each with parameter options, can be simulated. Queuing models are generally described in terms of the distribution of request occurrences (arrival distribution), the distribution of action occurrences (service distribution), and the number of action facilities (servers). Specifically, the times between request occurrences are taken to be independent and identically distributed realizations from the arrival distribution. The duration of each action occurrence is also taken to be an independent and identically distributed realization from the service distribution. Variations on these assumptions occur for some models and are described in Section A.

The abbreviation G/G/C describes a queuing system in which the arrival and service distributions are general (unspecified) and there are C servers.

The five options in this simulation are:

- (1) G/G/C
- (2) G/G/C with losses
- (3) G/G/C with feedback



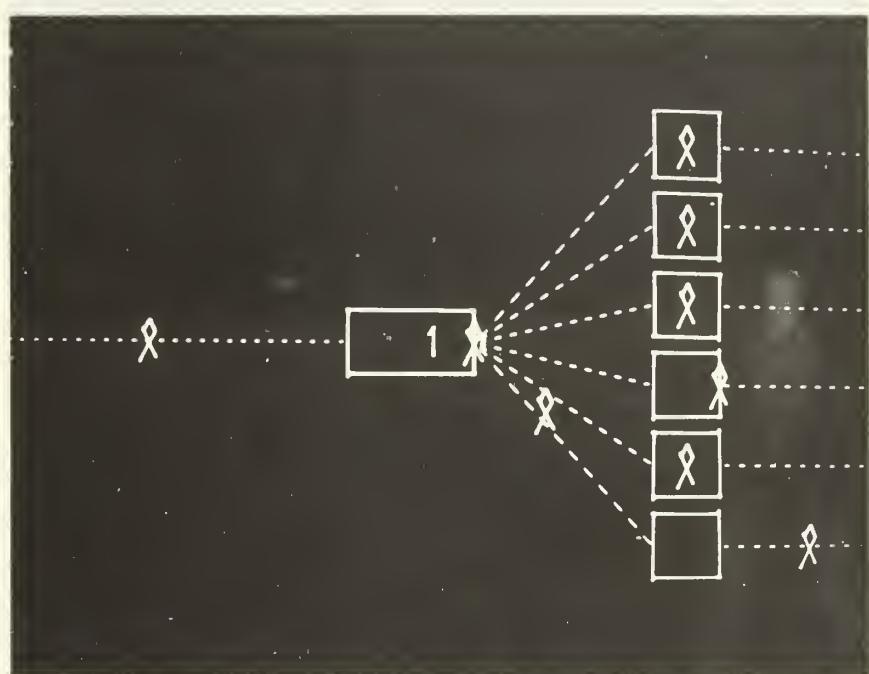
(4) G/G/C with finite source

(5) Two simultaneous G/G/C queues for comparison.

The specific choices of arrival distribution and service distribution which can be simulated are fully described in Section B.

#### A. THE QUEUING MODELS

##### 1. G/G/C



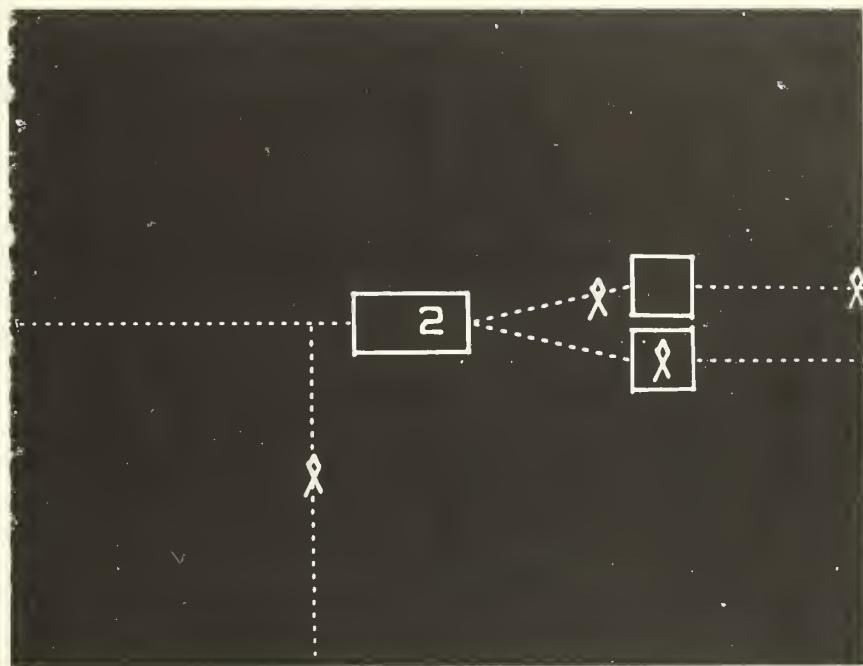
Picture 1

This is the simplest of the queuing models considered here. Each arrival waits in the queue until a server is available, enters service for a calculated period of time, then departs the system.

This model may be simulated with from one to six servers.



## 2. G/G/C with Losses

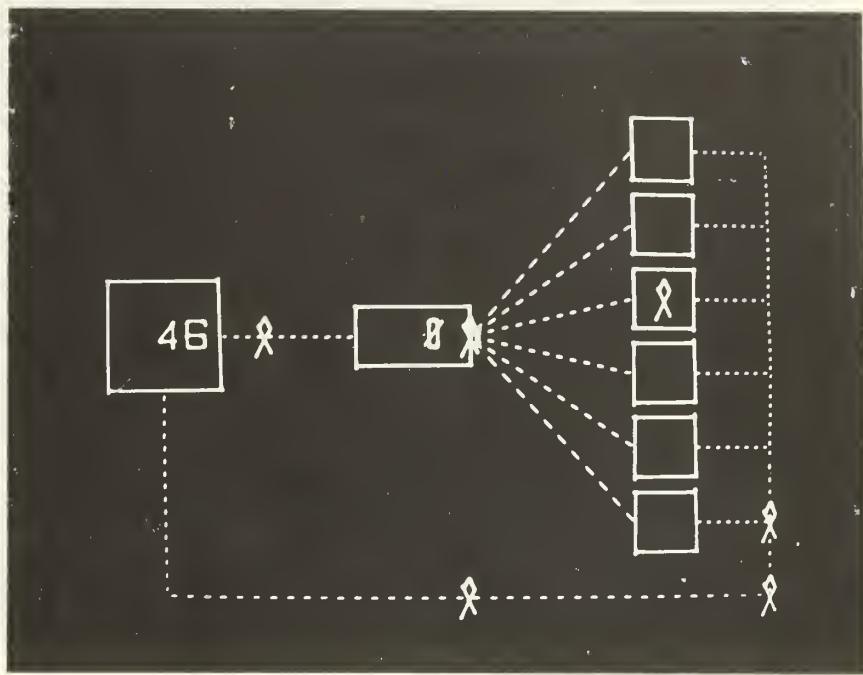


Picture 2

Consider making a long distance telephone call on Mother's Day. If all the trunk lines are busy, your request for service will be rejected. The G/G/C model with losses reflects this by rejecting arrivals who find the queue full. The queue limit used is specified as an input parameter. Note that rejected arrivals are lost and are not assumed to return later. The fraction of all arrivals who are served (not rejected) is recorded for subsequent display.



### 3. G/G/C with Finite Source



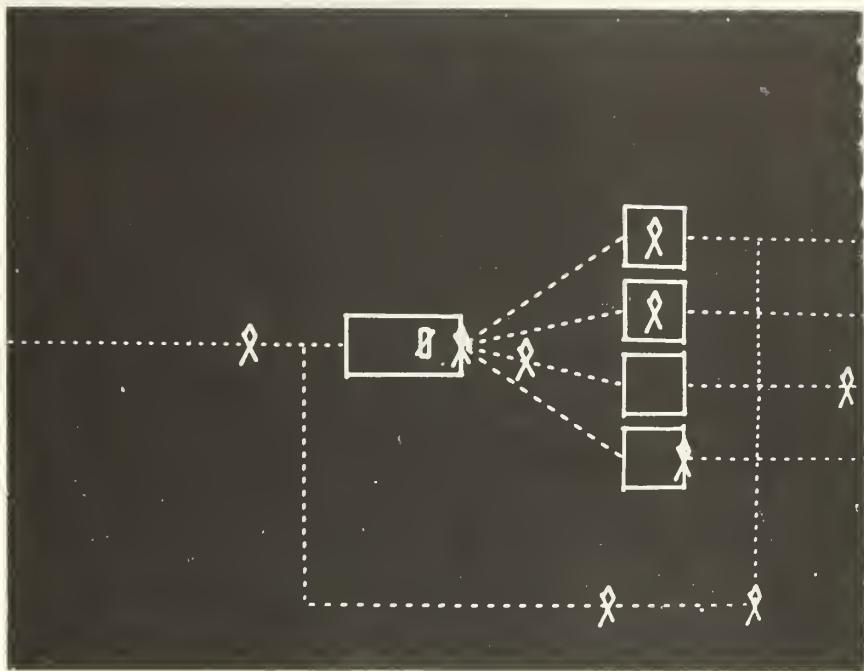
Picture 3

Consider a closed-circuit educational television system at a school. The finite population consists of all the televisions in the system. When one fails, it enters the queue to be serviced. After repair, it re-enters the population pending the next failure.

This situation is modeled by assuming that each action source, upon entering the idle population, is assigned a waiting time from the arrival distribution. When this time expires, the source requests service, joining the queue if necessary. As each action source terminates its service, it returns to the idle population. The user specifies the number of servers (one to six) and the size of the population.



#### 4. G/G/C with Feedback to the Queue

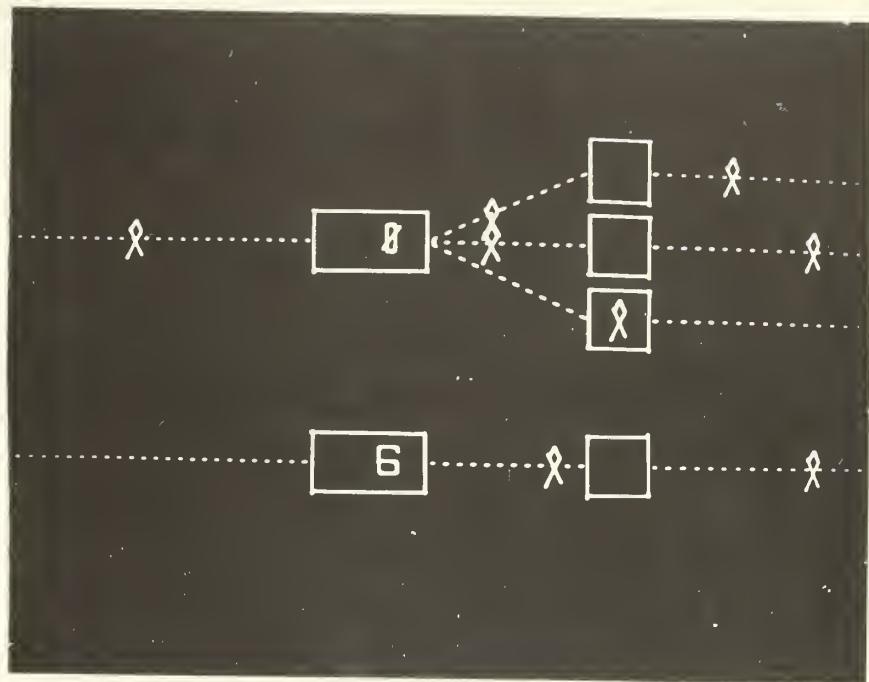


Picture 4

This is an extension of the G/G/C model, with from one to six servers, allowing customers leaving service to depart the system or to return to the queue. At each service completion, the customer leaves the system with probability  $P$  or returns to the queue with probability  $1-P$ . The user specifies  $P$  as an input option.



## 5. Two Simultaneous G/G/C Queues



Picture 5

This option allows comparison of two standard G/G/C queues, each with up to three servers. All parameters of each system are specified separately, and, if desired, identical arrival streams may be utilized. This allows comparisons between different arrival and/or service distributions and associated parameters, and between a different number of servers. Also, the effect of the various queuing disciplines can be compared.

### B. ARRIVAL/SERVICE DISTRIBUTIONS AND QUEUE DISCIPLINES

As well as options already described, the following arrival and service distributions are available to the user of the program. Any of the following probability



distributions can be used as an arrival or service distribution.

### 1. Distributions

#### a. K - ERLANG (exponential when K=1)

$$F(t) = 1 - \sum_{J=0}^{K-1} \frac{(\beta t)^J}{J!} e^{-\beta t}$$

$$\text{RATE } \lambda = \frac{\beta}{K}$$

$$CV = \frac{1}{\sqrt{K}}$$

Input parameters are:

(1) K

(2) Rate - in terms of mean number of events per minute.

#### b. Hyperexponential

$$F(t) = p(1-e^{-\lambda_1 t}) + q(1-e^{-\lambda_2 t})$$

where:

$$(1) p + q = 1$$

$$(2) CV = \frac{1}{\sqrt{p}}$$

(3)  $\lambda$  = Rate is specified and then

$\lambda_1, \lambda_2$  chosen to satisfy--

$$\lambda_2 = (2 + \sqrt{2})\lambda$$



$$\lambda_1 = \left( \frac{p(2 + \sqrt{2})}{2 + \sqrt{2} - q} \right) \lambda$$

Input parameters are:

- (1) Rate, specified in terms of mean number of events per minute.
- (2) Coefficient of variation (must be greater than 1.0)

c. Degenerate (Fixed Rate)

$$F(t) = \begin{cases} 0 & t < 1/\lambda \\ 1 & t \geq 1/\lambda \end{cases}$$

Input parameter:

- (1) Rate  $\lambda$  specified in terms of mean number of events per minute.

2. Queuing Disciplines

For comparison purposes, there are four queuing disciplines to choose from. These determine the priority by which members of the queue will enter service.

a. FIFO - the first into queue is the first out.

Over all disciplines which are independent of service times, this one minimizes the variance of delay.

b. LIFO - the last into the queue is the first out. Some systems, by their nature, function in this manner.

c. SSTF - the queue member with shortest service time is served first; this minimizes the mean delay over



all other disciplines, but the variance of delay is greater than that of FIFO.

d. RANDOM DRAW - applies to a system in which insufficient information is known to implement any other discipline.

### 3. Length of the Simulation

The period of simulation is specified in minutes by the user. All queues are empty and servers idle at the start of the simulation.



### III. THE METHOD

#### A. THE COMPUTER SIMULATION

The computer simulation is a modified next event type. There are two types of events, arrivals to the queue and a completion of service. With the first there will be an associated entry to service if there is an idle server, and with the last an associated entry to service if the queue is not empty.

The simulation is begun with the generation of the first arrival time and its associated service time. The times are generated via the transformation of random numbers to stochastic variates of the type specified. At the occurrence of an arrival, the next arrival time and associated service time are generated. When a server is available, a waiting member of the queue enters service with completion time equal to the sum of his assigned service time and the time he departs the queue. Thus with the clock starting at zero, a time history of events is created which tells what happened and when.

At the same time, the appropriate values are retained from which statistics of the simulation can later be calculated.

#### B. THE GRAPHICAL DISPLAY

At the completion of the simulation phase, the time history is depicted dynamically on the cathode ray tube



in real time. Pictures 1 to 5 show the form of the display, with the man-like images in motion along the appropriate path, and the number in the queue box showing the queue's current length.

The timing and motion of the display are controlled by the analog computer via the hybrid connections to the digital computer. The first connection is for the purpose of timing until the next event is to be initiated. In this case, the analog computer integrates from zero to a preset time value in volts, at the rate of one volt per second. The interrupt to the appropriate subroutine then occurs, and the analog computer is reset with the next time value, after which the integration is reinitiated.

The second connection causes the dynamic motion to occur by continually checking events in progress and causing the position of the corresponding images to be incremented appropriately; this interrogation and incrementation occurs approximately 10 times per second depending on the settings of the analog computer. Movement increments are between one-half inch and two inches (depending on the path), causing a motion of between five inches and 20 inches per second.



#### IV. STATISTICS FROM THE SIMULATION

STATISTICS					
	01	02		01	02
QT	.58	3.99	QC	.31	3.67
BT	2.31	.95	BC	2.27	2.33
LT	2.88	4.94	LC	2.58	6.88
DC	1.15	22.24	DV	5.65	111.86
SC	5.32	2.48	SV	26.59	11.27
WC	6.47	24.63	OV	3.67	38.14
DM	5.88	57.88	DM	3.88	9.88
DI	.58	.82	DS	.35	.81
DX	.46	.82	DL	.14	.76
NC	26	24	PL	.88	
ENTER 1. CONTINUE					
2. PARAMETER LISTING					

At the end of the simulation period, the following statistics are calculated and displayed, as shown above.

AT = time - average length of the queue

BT = time - average number of busy servers

LT = time - average number in the system = QT + BT

QC = customer - average length of the queue

BC = customer - average number of busy servers

LC = customer - average number in the system = QC + BC

DC = customer - average delay in the queue

SC = customer - average service time

WC = customer - average wait in the system = DC + SC



DM = maximum delay experienced by any customer

DO = probability that a customer's delay is greater than zero.

DX = probability that a customer's delay is greater than XD. (The value of XD is an input option.)

QM = maximum queue length reached during the simulation

QO = probability that the queue length will be greater than zero.

QL = probability that queue length will be greater than LQ (the value of LQ is an input option).

DV = standard deviation of the delay in queue

SV = standard deviation of the service times

QV = standard deviation of the queue length

NC = total number of arrivals

PS = percentage not lost (losses model only)



## V. PARAMETER INPUT

ITYPE	5	RUNTIME	1.88
IC	3	IC1	1
	1	2	3
IDIST	1	1	1
KK	1	1	1
RATE	38.88	38.88	14.88
CV	1.88	1.88	1.88
IDSPLN	1	1	
LO	1	MAXD	1
XD	1.88	NPOP	1
IDUAL	1	P	1.88
ENTER 1. CONTINUE			
2. STATISTICS			

Picture 7

Parameters can be input by the card reader or by the teletype terminal. In either case, the namelist form is used, where the entry form is NAME = X, NAME being the variable title and X being the value being assigned. For real variables, the decimal point must be entered. Namelist input is terminated with a \*.

### A. SAMPLE INPUT CARD

---

IDEV = 2, ITYPE = 3, RATE(2) = 3.0 \*



## B. NAMELIST INPUT VIA THE GRAPHICS TERMINAL

If this mode of entry is specified, one variable at a time is entered, followed by a carriage return. Entries are displayed, as entered, on the graphics screen. Termination occurs with the entry of a \* followed by a carriage return.

## C. INPUT VARIABLES

The following conventions are employed for specifying the options and parameters to be used.

ITYPE = 1 gives G/G/C

2 gives G/G/C with losses

3 gives G/G/C with finite source

4 gives G/G/C with feedback

5 gives two simultaneous G/G/C queues

RUNTIME = desired length of simulation (minutes)

IC = number of servers for queue 1

IC1 = number of servers for queue 2 (only applicable if

ITYPE = 5 is chosen)

IDIST(I) = 1. K - ERLANG

2. HYPEREXPONENTIAL

3. DEGENERATE

KK(I) = K in K-ERLANG (use K=1 for exponential)

RATE(I) = Rate in number per minute

CV(I) = Coefficient of variation (specify for IDIST(I) = 2 only. This variable is computed otherwise.)



where I = 1 for arrivals to queue 1 (except finite model)  
2 for arrivals to queue 2 or finite population  
idle parameters  
3 for service for queue 1 (except finite model)  
4 for service for queue 2 or finite population  
service parameters

IDSPLN(I) = 1. FIFO  
2. LIFO  
3. Shortest service time first  
4. Random draw

where I = 1 for queue 1  
2 for queue 2

LQ = parameter in the statistic  $P(Q > LQ)$

XD = parameter in the statistic  $P(\text{wait} > XD)$

IDUAL = 0. Independent arrival streams  
1. Identical arrival streams  
(applicable to ITYPE = 5 only)

MAXQ = maximum queue size (for G/G/C with losses only)

NPOP = population size (for finite source option only)

P = completion probability (for feedback option only)



## VI. THE PROGRAM

The program requires the use of the XDS 9300 digital computer, the CI 5000 analog computer, and a graphics terminal using the GATED monitor.

The analog computer wiring diagram necessary to drive the display sequence appears in APPENDIX A. Before beginning, set Delay Flop Switches DFO0, DFO1, and DFO2 to .1 MS and set counter F00 to 1 and F04 to 0. Then select RUN and DIGITAL COMPUTER on the analog control panel. This provides for movement interrupts to occur at a rate of 100/11 hertz. If desired, the timing values will be displayed on the RATIO METER by selecting TRUNK 420 prior to DIGITAL COMPUTER.

The program consists of a FORTRAN program and a METASYMBOL 9300 program. Because the history is buffered out to the device assigned the logical unit value 7, an additional control card is needed.

Job cards for deck run should be:

```
¬JOB
¬AGT
¬ASSIGN 7=DF1A
¬FORTRAN LS,GO
```

FORTRAN DECK

```
¬META 9300 SI,LO,GO
```

META-SYMBOL DECK



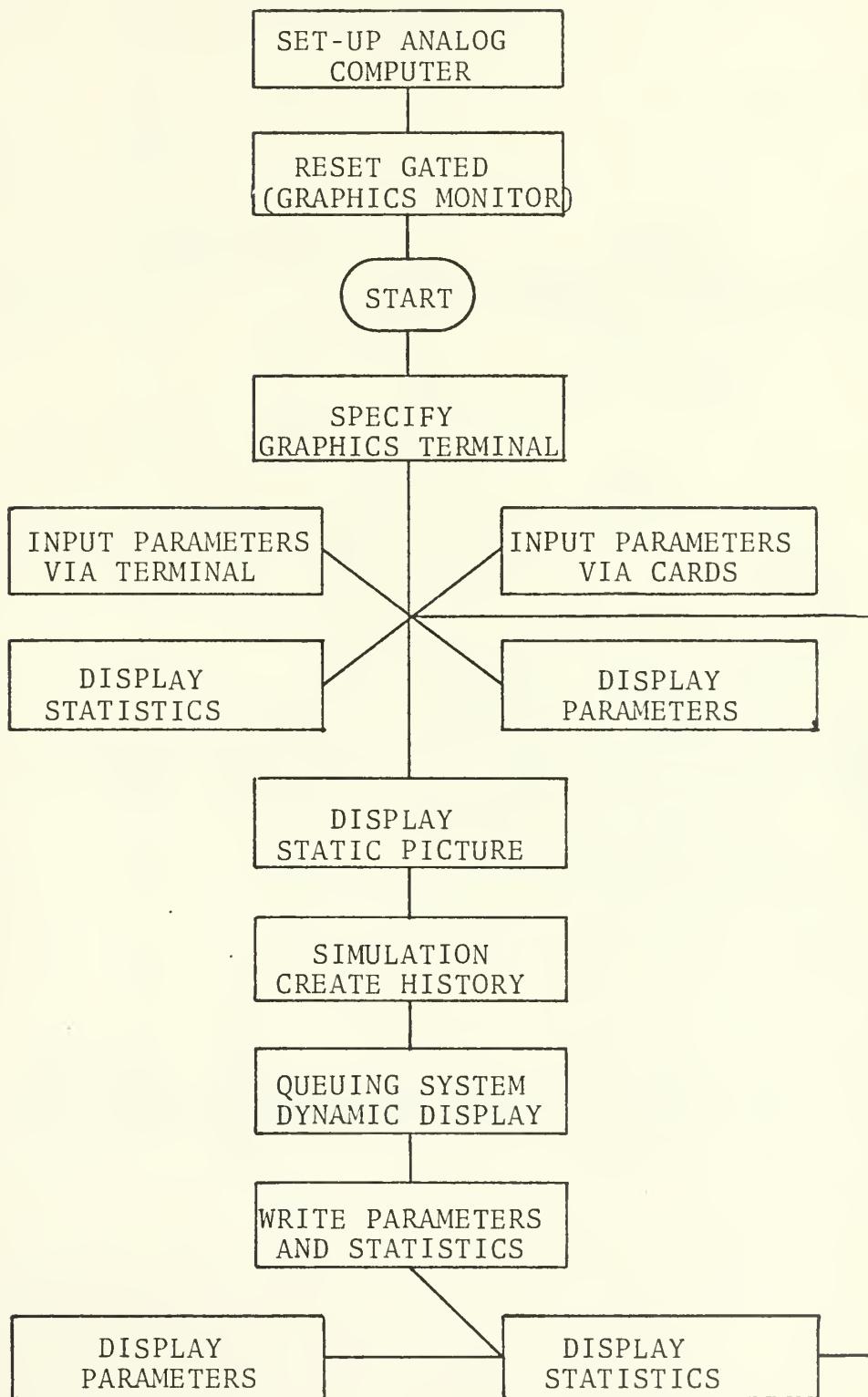
¬EOF

¬LOAD XR,MAP

¬DATA

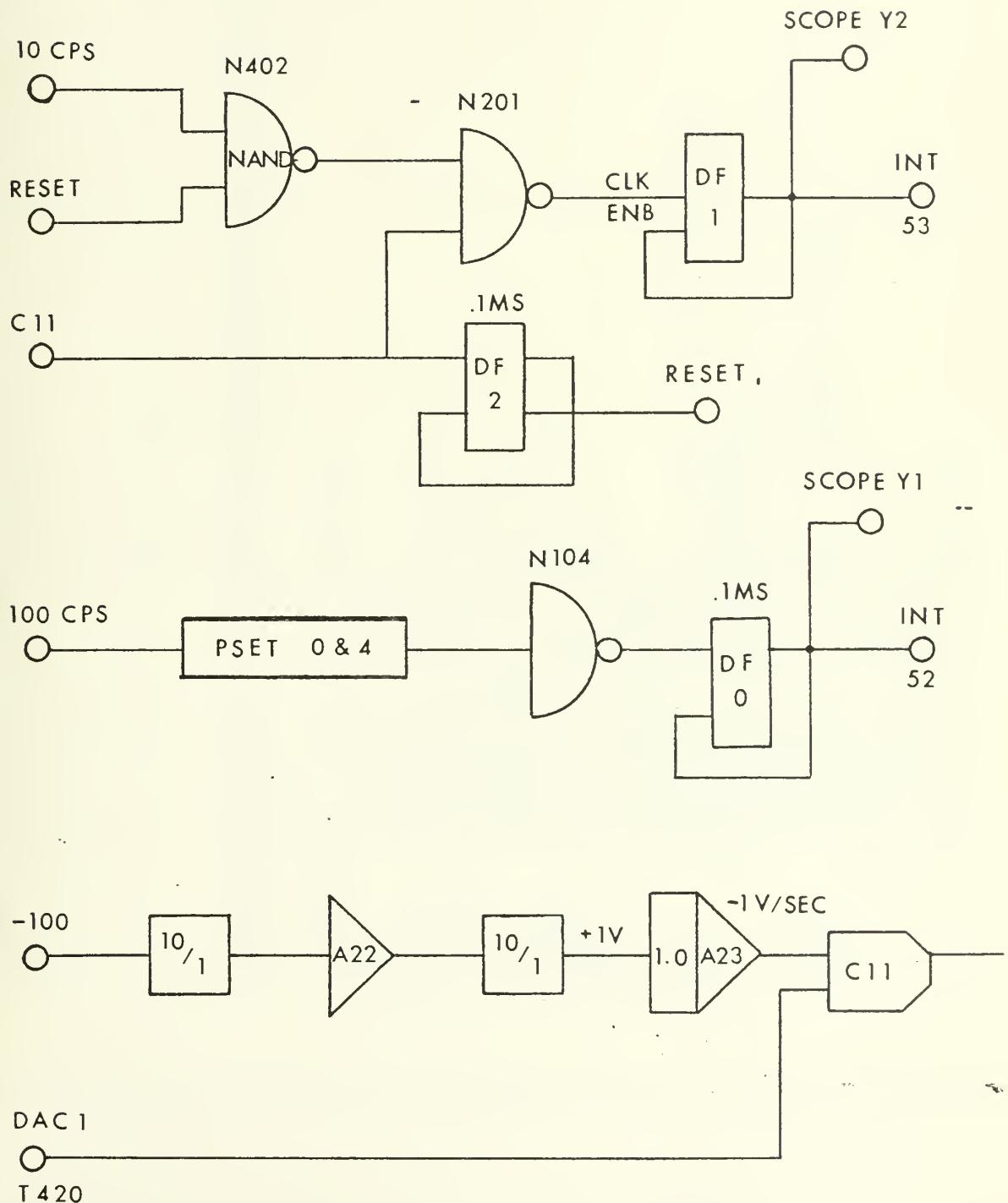


PROGRAM STEPS





APPENDIX A  
ANALOG SCHEMATIC





REPORT NO. SIMULATION OF QUEUEING MODELS WITH DYNAMIC DISPLAY

C. M. RIDDELL  
MARCH 1973

DICTIONARY OF EXOGENOUS VARIABLES

CHNG	- INCREMENT VALUES
CLOCK	- HISTORY TIMES
CLTIME	- SERVICE TIME SUM
CLTIME0	- (SERVICE TIME) * 2 SUM
CLTIMEQ	- (QUEUE LENGTH) * 2 SUM
CUSTIM	- QUEUE LENGTH SUM
CV	- COEFFICIENT OF VARIATION
DELAY	- QUEUE WAIT SUM
DELAYO	- DELAYS GREATER THAN 0
DELAYSQ	- (QUEUE WAIT) * 2 SUM
DELAYX	- DELAYS GREATER THAN XD
END	- MOVEMENT END VALUES
FTIME	- NEXT EVENT TIMES
I3	- HISTORY POINTER
IBUFF	- HISTORY BUFFER
IBUSY	- BUSY? 0 - NO: 1 - YES: 2 - NOT APPLICABLE
IC	- NUMBER OF SERVERS 1
IC1	- NUMBER OF SERVERS 2
IDEV	- SPECIFIES GRAPHICS TERMINAL
IDIST	- DISTRIBUTION TYPE
IDSPLN	- QUEUE DISCIPLINE
IDUAL	- IDENTICAL ARRIVALS? 0 - NO: 1 - YES
IFDBK	- FEEDBACK? 0 - NO: 1 - YES
IFLAG	- DYNAMIC IMAGE FLAGS
ILIN	- TEXT LINE NUMBER
IMAN	- DYNAMIC IMAGES
IMD	- MOVE-DRAW INDICATORS
IPOS	- TEXT STARTING POSITION
ISTOT	- ARRIVAL BUSY SERVER SUM



```

-- ITOT INTERMEDIATE HISTORY COUNTER
-- ITPOP POPULATION SIZE TEXT WORD
-- ITQ QUEUE LENGTH TEXT WORDS
-- ITXA MAIN CHOICE PAGE TEXT WORD
-- ITTYPE MODEL TYPE
-- ITX1-IX4 RANDOM NUMBER SEEDS
-- KIM HISTORI IMAGE POINTER
-- KK (ERLANG)
-- KPO VECTORS POINTERS FOR TIM
-- KLINE LENGTH OF QUEUES
-- KLOG HISTORI LOGIC POINTER
-- KQ LOWER LIMIT QUEUE LENGTH VALUE
-- KTOT ARRIVAL QUEUE SUM
-- MAXLN MAXIMUM QUEUE VALUE REACHED
-- MAXQ DESIRED MAX QUEUE LENGTH
-- MD MOVE-DRAW INDICATORS
-- XMDELAY MAXIMUM DELAY IN QUEUE
-- NCUST ARRIVAL COUNTERS OF FEEDBACKS
-- NFDBK FINITE POPULATIONS SIZE
-- NPOP LENGTH OF QUEUES
-- NQ NUMBER OF STATIC IMAGES
-- NSIM TOTAL NUMBER OF HISTORY ENTRIES
-- NTOT BLANK
-- NULL PROBABILITY
-- PROBO QUEUE GREATER THAN O SUM
-- PROBL QUEUE GREATER THAN LQ SUM
-- PSTIME PREVIOUS STATE CHANGE TIME
-- PTIME PREVIOUS QUEUE CHANGE TIME
-- RATE RATE
-- RUNTIME LENGTH OF SIMULATION
-- TIME STORED TIME
-- TIME CURRENT TIME
-- TIME LOWER LIMIT TIME VALUE
-- XD SCREEN X COORDINATES
-- XX DYNAMIC IMAGE X COORDINATES STARTING POSITIONS
-- YY SCREEN Y COORDINATES
-- YY DYNAMIC IMAGE Y COORDINATES STARTING POSITIONS
-- ZY DYNAMIC IMAGE Z COORDINATES
-- ZY DYNAMIC IMAGE Z COORDINATES STARTING POSITIONS

```



```

DIMENSION IBDIFF(300),ITDIR(16)
DIMENSION ITXA(6)
REAL LT,LC
COMMON /SET3/X(14),Y(25),MD(6),IC,IC1,NULL
COMMON /SET5/IDEV,IER,KIM(100),LOG(100)
COMMON /SET6/CLOCK(100),END(5,22),ITFLAG(22),ITOT,ITPOP(2),XX(5,22),
COMMON /SET8/CHNG(10),NSIM(2),ITQ(2),NPOP,ITPOP(2),XX(5,22),
-ILN(2),IMAN(7,22),NQ(2),IMD(5),
-XXX(5,22),YY(5,22),YYY(5,22),
-COMMON /SET9/IDIST(4),RATE(4),KK(4),CV(4),NFDBK,IDSPLN(2),
-IDUAL,IBUSY(6),KP(2),LINE(2),TIME,PTIME,PTIME(2),RUNTIME,IX1,
-IX2,IX3,IX4,XD,LQ,MAXQ,P
COMMON /SET10/NTOT,ITYPE
COMMON /SET11/NCUST(2),MAXLN(2),DELAY(2),XMDELAY(2),
COMMON /SET12/NCUST(2),CLTIME(2),CLTIMSQ(2),CTIMSQ(2),
-DELAYO(2),DELAYX(2),CLTIME(2),CLTIMSQ(2),CTIMSQ(2),
-PROB0(2),PROBL(2),LTOT(2),ISTOT(2),
COMMON /STAT/QT(2),BT(2),DO(2),DX(2),QC(2),BC(2),LC(2),
-DC(2),SC(2),WC(2),DW(2),PL,NC(2),
-QL(2),DV(2),SV(2),PL,NC(2),
COMMON /SETX/A1,A2,A3,A4,G1,G2,63,64,S1,S2,S3,S4,S5,S6,S7,S8
EQUIVALENCE (IBUFF(1),CLOCK(1),(IBUFF(201),KIM(1)),
-(IBUFF(301),LOG(1))
DATA ITQ(2,1)*ITQ(2,2)*ITPOP(2),NULL/7777777777B/
DATA IX1,IX2,IX3,IX4,55403,76941,10799,33999,
DATA X/-1.08,-.94,-.81,-.74,-.66,-.58,-.51,-.26,-.15,.05,
-74*94*1.34/
DATA Y/57*48*4*4*36*28*2*16*13*08*06*0*02*04,
-1*12*17*2*2*24*32*44*52*56*72*71*0/
DATA ITYPE,IC,IC1,IDIST,KK,IDSPLN,LQ,IDUAL,MAXQ,NPOP,i/
DATA RATE,CV,RUNTIME,XD,P/1.0/
DATA MD/6*0/,IMD/0.4*1/

```

CCC CCC CCC

ESTABLISH NAMELIST VARIABLES (FOR INPUT)  
 NAMELIST ITYPE,IC,IC1,IDIST,KK,RATE,CV,IDSPLN,RUNTIME,  
 -LQ,XD,IDUAL,MAXQ,NPOP,IPRINT

FORMAT FOR MAIN CHOICE PAGE

```

101 FORMAT('1. INPUT BY CARDS
102 FORMAT('2. INPUT BY TERMINAL')
103 FORMAT('ENTER CHOICE THEN RETURN')
104 FORMAT('3. PARAMETER LISTING')
105 FORMAT('4. RUN',17X)

```

MAKE THE HYBRID CONNECTIONS



```

C 52 INTERRUPT CAUSES MOTION
C 53 INTERRUPT TIMES EVENTS
C CONNECT (52,FLAGS)
C CONNECT (53,SETFLAGS)
C
C DEFAULTS: NO HISTORY OUTPUT; GRAPHICS TERMINAL 1
C
C IPRINT=0; IDEV=1
C OUTPUT(101) *TYPE IDEV=2 * AND A C/R IF AGT-2 IS TO BE USED ELSE T
C -TYPE * AND A C/R
C INPUT(101)
C
C SET THE STATIC IMAGE COORDINATES
C
C CALL PACK
C
C INITIALIZE THE GRAPHICS AND TEXT DIRECTORIES
C
C CALL DGINIT (IDEV,IGDIR,30,IER)
C CALL DTINIT (IDEV,ITDIR,16,IER)
C
C PRESENT MAIN CHOICE PAGE
C
C 110 ENCODE(24,101,ITXA)
C CALL TEXT0 (IDEV,ITXA,6,15,25,2,3,IER)
C ENCODE(24,102,ITXA)
C CALL TEXT0 (IDEV,ITXA,6,17,25,2,3,IER)
C ENCODE(24,104,ITXA)
C CALL TEXT0 (IDEV,ITXA,6,19,25,2,3,IER)
C ENCODE(24,105,ITXA)
C CALL TEXT0 (IDEV,ITXA,6,21,25,2,3,IER)
C ENCODE(24,103,ITXA)
C CALL TEXT0 (IDEV,ITXA,6,24,25,2,3,IER)
C CALL TEXTR (IDEV,NULL,1,26,47,2,3,IER)
C
C WAIT FOR REPLY
C
C 100 IF(MOD(ITDIR(6),8)=0) GO TO 100
C CALL TEXTI (IDEV,I,1,26,47,IER)
C
C LOGICAL RIGHT SHIFT 18 BITS (3 CHARACTERS) DECODES REPLY
C I=LRS(I,18)
C
C CALL DTINIT(IDEV,ITDIR,16,IER)
C
C GET CARD INPUT

```







```

IF(I•LE•IC) IBUSY(I)=2
CONTINUE
DO TO 7
IF(I•LE•IC) IBUSY(I)=2
IF(I•LE•IC1) IBUSY(I)=0: GO TO 5
IF(I•LE•IC1) IBUSY(I+3)=2
CONTINUE
NFDBK=IFDBK=ITOT=NTOT=TIME=POSTIME=0
DO 8 I=1•22
IFLAG(I)=0
DO 15 I=1•2
KP(I)=LINE(I)=NQ(I)=NCUST(I)=MAXLN(I)=DELAY(I)=DELAYSQ(I)=
-XMDELAY(I)=DELAY(I)=CLTIME(I)=CLTMSQ(I)=
-PTIMF(I)=CUSTIM(I)=CTIMSQ(I)=PROBL(I)=
-LTOT(I)=ISTOT(I)=0
-IF(I•TYPE•EQ•5) GO TO 50
A1=A2=1: A3=A4=12
G1=G2=10: G3=G4=12
S1=S2=12: S3=S4=9: S5=19: S6=5: S7=22: S8=2
CHNG(5)=-.0333: CHNG(6)=.0333: CHNG(7)=-.1
CHNG(8)=.1: CHNG(9)=-.1666: CHNG(10)=.1666
ILN(1)=ILN(2)=22: IPOS(1)=41: IPOS(2)=16
GO TO (10,20,30,40) ITYPE
SETUP G/G/C
CALL GGC
DO 11 I=1•4
ENDD(5,I)=X(9)
DO 12 I=5•10
ENDD(1,I)=X(12)
DO 13 I=11•22
ENDD(5,I)=X(14)+.1
CHNG(1)=.225: CHNG(2)=.17: CHNG(3)=.2
NSIM=5
GO TO 85
SETUP G/G/C WITH LOSSES
CALL LOSSES
ENDD(5,1)=ENDD(5,3)=ENDD(1,2)=ENDD(1,4)=X(9)
ENDD(2,2)=ENDD(2,4)=Y(25)
DO 22 I=5•10
ENDD(1,I)=X(12)
DO 21 I=11•22

```



```

21 ENDD(5,1)=X(14)+.1 CHNG(2)=.17: CHNG(3)=.205
NSIM=6
GO TO 85

C C SETUP G/G/C WITH FINITE SOURCE
30 CALL FINITE
    A1=A2=7
    DO 31 I=1,4
31 ENDD(5,1)=X(9)+.03
    DO 32 I=5,10
32 ENDD(1,1)=X(12)
    DO 33 I=11,22
33 ENDD(1,1)=X(13)+.1
    ENDD(2,1)=Y(24)-.1
    ENDD(3,1)=X(5)
    ENDD(4,1)=Y(16)
    CHNG(1)=.12: CHNG(2)=.17: CHNG(3)=.2: CHNG(4)=.275
    NSIM=7
GO TO 85

C C SETUP G/G/C WITH FEEDBACK TO THE QUEUE
40 CALL FEEDBK
    DO 41 I=1,4
41 FNDD(5,1)=X(9)
    DO 42 I=5,10
42 ENDD(1,1)=X(12)
    DO 43 I=11,22
43 ENDD(1,1)=X(13)+.1
    ENDD(2,1)=Y(24)-.1
    ENDD(3,1)=X(8)
    ENDD(4,1)=Y(12)
    ENDD(5,1)=X(14)+.1
    CHNG(1)=.225: CHNG(2)=.17: CHNG(3)=.2: CHNG(4)=.35
    NSIM=7
GO TO 85

C C SETUP TWO G/G/C QUEUES
50 CALL TWOQGC
    ILN(1)=17: ILN(2)=27: IPOS(1)=IPOS(2)=41
    A1=A2=1: A3=5: A4=19
    G1=G2=10: G3=5: G4=19
    S1=S2=12: S3=5: S5=2: S6=19: S7=22: S8=15
    DO 51 I=1,4
51 ENDD(5,1)=X(9)

```



```

DO 52 I=5,10
52 ENDD(1,I)=X(12)
DO 53 I=11,22
53 ENDD(5,I)=X(14)+1
CHNG(1)=2.25; CHNG(2)=.17; CHNG(3)=.2
CHNG(5)=CHNG(8)=0.0
CHNG(6)=CHNG(9)=-.0666; CHNG(7)=CHNG(10)=.0666
NSIM=6

C SET COORDINATES FOR DYNAMIC IMAGES
C
85 CALL XFS
C BEGIN THE SIMULATION
C CALL SIMULATE
C PRINT THE HISTORY?
IF(IPRINT.EQ.0) GO TO 90
DO 86 J=1,IPRINT
86 OUTPUT(6) CLOCK(J),KIM(J),LOG(J)
C BEGIN THE DYNAMIC GRAPHICS DISPLAY
C
90 CALL DRIVER
C LIST STATISTICS
C
C OUTPUT(6) ITYPE,RUNTIME,IC,IC1,IDI ST,KK,RATE,CV,
-IDSPLN,LQ,XD,IDL,MAXQ,NPOP,P
-OUTPUT(6) QT,BT,LT,QC,BC,LC,DC,SC,WC,DM,DO,DX,QM,QO,QL,
-DV,SV,QV,PL,NC
CALL DGINIT(IDEV,IGDIR,30,IER)
CALL DTINIT(IDEV,ITDIR,i6,IER)
C CONVERT SECONDS TO MINUTES
C
DO 95 I=1,4
95 RATE(I)=RATE(I)*60.
RUNTIME=RUNTIME/60.
C DISPLAY STATISTICS
C
91 CALL STATISTICS
CALL TEXTTR(IDEV,NULL,1,35,11,2,3,IER)
150 IF(MOD(ITDIR(15),8).EQ.0) GO TO 150
150 CALL TEXTI(IDEV,I,1,35,11,IER)

```



```

CALL DTINIT(IDEV,ITDIR,16,IER)
I=LRS(1,18)
IF(I.EQ.1) GO TO 110
IF(I.NE.2) GO TO 91
C DISPLAY PARAMETERS?
C
 93 CALL PARAMETERS
  CALL TEXTR(IDEV,NULL,1,34,9,2,3,IER)
175 IF(MOD(ITDIR(14),8).EQ.0) GO TO 175
  CALL TEXTI(IDEV,I,1,34,9,IER)
  CALL DTINIT(IDEV,ITDIR,16,IER)
I=LRS(1,18)
IF(I.EQ.2) GO TO 91
C RETURN FOR ANOTHER RUN
C
  GO TO 110
END

```



## SUBROUTINE SIMULATE

PURPOSE: TO CREATE THE TIME HISTORY OF EVENTS  
AND GATHER STATISTICS

SUBROUTINES USED: ARRIV: BEGSRV: ENDSRV: ACCUM: TIMES

```

SUBROUTINE SIMULATE
DIMENSION IBUFF(400)
REAL LT,LC
COMMON /SET6/ CLOCK(100),KIM(100),LOG(100)
COMMON /SET7/ ETIME(9),I3,K1,TIM(115,4)
COMMON /SET8/ CHNG(10),ENDD(5,22),IFLAG(22),ITOT(2),IPOS(2),
-ILN(2),IMAN(7,22),NSIM(2),EQ(2),ITOT(2),NPOT(2),ITPOP(2),
-XXX(5,22),YY(5,22),IMD(5)
COMMON /SET9/ IDIST(4),RATE(4),KK(4),CV(4),NFDBK,IFDBK,IDSPLN(2),
-INDUAL,IBUSY(6),KP(2),LINE(2),TIME,PSTIME,PTIME,PRUNTIME,IX1,
-IX2,IX3,IX4,XD,LQ,MAXQ,P
COMMON /SET10/ NTOT,ITYPE
COMMON /SET11/ NCUST(2),MAXLN(2),CLTIME(2),CLTMSQ(2),XMDELAY(2),
-DELAY0(2),DELAYX(2),CLTMSQ(2),CUSTIM(2),CTIMSQ(2),
-PROB0(2),PROBL(2),LTOT(2),ISTOT(2)
COMMON /STAT/ QT(2),BT(2),LT(2),OC(2),BC(2),LC(2),
-DC(2),SC(2),WC(2),DM(2),DX(2),Q(2),Q0(2),
-QL(2),DV(2),SV(2),PLNC(2)
EQUIVALENCE (IBUFF(1),CLOCK(1)),(IBUFF(201),KIM(1)),
-(IBUFF(301),LOG(1))
-13=1

```

SEVEN TIMES 10 LARGE VALUE

```

DO 10 K=1,8
10 ETIME(K)=99999.
IF(ITYPE.NE.3) GO TO 4
I=2

```

## IDLE TIMES FOR FINITE POPULATION

```

DO 3 J=1,NPOP
  CALL TIMES
  TIM(KP(1)+1,1)=TIM(KP(2),2)
  TIM(KP(1)+1,3)=TIM(KP(2),4)
  3

```



```

C KP(2)=KP(2)-1
C SET FIRST ARRIVAL FROM FINITE SOURCE
C ETIME(7)=TIME(1)+1.1
C GO TO 5
C GET FIRST ARRIVAL TIME
C 4 I=1: CALL TIMES; ETIME(7)=TIME(1,1)
C IF(I TYPE.EQ.5) I=2: CALL TIMES; ETIME(8)=TIME(1,2)
C SET LENGTH OF SIMULATION
C 5 ETIME(9)=RUNTIME
C DETERMINE NEXT EVENT
C 11 TIME=ETIME(1)
C K1=1
C DO 15 K=2,9
C IF(ETIME(K).LT.TIME) TIME=ETIME(K): K1=K
C 15 CONTINUE
C SEPARATE END-OF-SERVICE
C IF(K1.GT.6) GO TO 20
C I=1
C IF(I TYPE.GE.5 .AND. K1.GE.4) I=2
C CALL ENDSRV: GO TO 11
C SEPARATE ARRIVALS
C 20 IF(K1.LE.8) I=K1-6: CALL ARRIV: GO TO 11
C DONE: COMPLETE HISTORY
C CALL BUFFEROUT(7,I,BUFF,400,IND)
C 21 IF(IND.EQ.1) GO TO 21
C GO TO (21,23,22,22) IND
C 22 OUTPUT(101,BUFFERING,ERROR 5)
C 23 REWIND 7
C I=1
C COMPLETE TIME STATISTICS
C 50 CALL ACCUM

```



C

## CALCULATE FINAL STATISTICAL VALUES

```

TCUST=NCUST(1)
IF(ITYPE.EQ.2) TCUST=NCUST(1)+NCUST(2);
-PL=NCUST(1)/TCUST; NCUST(1)=TCUST
DO 55 J=1*KP(1)
WAIT=TIME-TIM(j,1)
DELAY(1)=DELAY(j)+WAIT
DELAY(1)=DELAY0(1)+1
IF(WAIT*GT*XD) DELAYX(1)=DELAYX(1)+1
SC(1)=CLTIME(1)/(TCUST-LINE(1))
SV(1)=SQR(CLTIME(1)/(TCUST-LINE(1))-SC(1)*#2)
K=1; M1=1
IF(ITYPE.NE.5) M2=6; GO TO 59
IF(I.EQ.1) M2=6; K=2
M1=4; J=M1*M2
59 DO 60
IF(ETIME(j)*EQ.99999.) GO TO 60
CLTIME(K)=CLTIME(K)-(ETIME(j)-TIME)
60 CONTINUE
QT(1)=CUSTIN(1)/TIME
BT(1)=CLTIME(1)/TIME
LT(1)=QT(1)+BT(1)
QC(1)=LTOT(1)/TCUST
BC(1)=ISCT(1)+RC(1)/TCUST
LC(1)=QC(1)+DC(1)+SC(1)
DC(1)=DC(1)+SC(1)
WC(1)=XMDELAY(1)
DM(1)=XMDELAY(1)/TCUST
DO(1)=DELAY0(1)/TCUST
DX(1)=DELAYX(1)/TCUST
QM(1)=MAXLN(1)
QO(1)=PRCBO(1)/TIME
QL(1)=PROBL(1)/TIME
DV(1)=SQR(CLTIME(1)-SC(1)*#2)
QV(1)=SQR(CLTIME(1)-SC(1)*#2)
NC(1)=NCUST(1)
IF(ITYPE.NE.2) PL=0.0
IF(ITYPE.NE.5) GO TO 70
IF(I.EQ.1) I=2; GO TO 50
GO TO 80
70 QT(2)=BT(2)=LT(2)=QC(2)=BC(2)=LC(2)=DC(2)=SC(2)=
-WC(2)=DV(2)=SV(2)=QV(2)=DM(2)=DO(2)=DX(2)=QM(2)=
-QO(2)=QL(2)=NC(2)=0
80 RETURN
END

```



## SUBROUTINE ARRIV

PURPOSE: RECORD AN ARRIVAL TO THE QUEUE IN THE HISTORY, GATHER STATISTICS, AND DETERMINE IF A SERVER IS AVAILABLE



```

C IF(LINE(1).LT.MAXQ) GO TO 5
C
C HISTORY ENTRY
C
C KIM(I3)=NO(2)
C
C ALTERNATE IMAGES
C
C IF(NO(2).EQ.2) NO(2)=4: GO TO 101
C NO(2)=2
C LOG(I3)=2
C I3=I3+1: NTOT=NTOT+1
C IF(I3.NE.101) GO TO 4
C CALL BUFFEROUT(7,1,IBUFFF,400,IND)
1 IF(IND.EQ.1) GO TO 1
2 GO TO (1,3,2,2) IND
2 OUTPUT(101) *BUFFERING ERROR 1
3 I3=1
3 NCUST(2)=NCUST(2)+1
4 PSTIME=TIME
KP(1)=KP(1)-1
GO TO 40
5 IF(ITYPE.GE.5 .OR. IF(J.EQ.1) J=2: GO TO 6
J=1
J=1

C ALTERNATE IMAGES
C
C 6 IF(NO(J).EQ.J) NO(J)=J+2: GO TO 6
C NO(J)=J

C HISTORY ENTRY
C
C 8 KIM(I3)=NO(J)
C CLOCK(I3)=TIME-PSTIME
C LOG(I3)=1
C I3=I3+1: NTOT=NTOT+1
C IF(I3.NE.101) GO TO 12
C CALL BUFFEROUT(7,1,IBUFFF,400,IND)
9 IF(IND.EQ.1) GO TO 9
GO TO (9,11,10,10) IND
10 OUTPUT(101) *BUFFERING ERROR 2
11 I3=1

C TIME STATISTICS

```



```

13 NCUST(I)=NCUST(I)+1
LINE(I)=LINE(I)+1
IF(IFDBK.EQ.1) IFDBK=0; GO TO 42
C C SERVER AVAILABLE?
C C
IF(LINE(I).GT.1) GO TO 40
LOG(I3)=7
IF(ITYPE.LE.4) GO TO 30
1F(I.EQ.2) GO TO 20
DO 15 K1=1,3
15 IF(IBUSY(K1).EQ.0) CALL BEGSRV; GO TO 40
GO TO 40
DO 21 K1=4,6
21 IF(IBUSY(K1).EQ.0) CALL BEGSRV; GO TO 40
GO TO 40
DO 31 K1=1,6
31 IF(IBUSY(K1).EQ.0) CALL BEGSRV; GO TO 40
40 IF(ITYPE.NE.3) GO TO 45
C C FINITE SOURCE EMPTY?
C C
IF(KP(2).EQ.0) ETIME(7)=99999.; GO TO 42
C C SWITCH NEXT ARRIVAL TO QUEUE HALF OF ARRAY
C C
TIM(KP(1)+1,1)=TIM(KP(2),2)
TIM(KP(1)+1,3)=TIM(KP(2),4)
KP(2)=KP(2)-1
GO TO 41
C C GET NEXT ARRIVAL TIMES
C C
45 CALL TIMES
45 ETIME(I+6)=TIM(KP(I)+1,I)
42 IF(LINE(I).GT.MAXLN(I)) MAXLN(I)=LINE(I)
END

```



ROUTINE BEGSRV  
PURPOSE: RECORD BEGINNING OF SERVICE: SELECTS QUEUE MEMBER  
AND SETS END OF SERVICE TIME

```

SUBROUTINE BEGSRV
DIMENSION IBUFF(400)
COMMON /SET6/ CLOCK(100), KIM(100), LOG(100)
COMMON /SET7/ ETIME(9), I13, K1, TIM(115,4)
COMMON /SET9/ IDIST(4), RATE(4), KK(4), CV(4)
- DUAL, IBUSY(6), KP(2), LINE(2), TIME, PTIME, PTIME, IX1,
- IX2, IX3, IX4, XD, LQ, MAXQ, P
COMMON /SET10/ NTOT, ITYPE
COMMON /SET11/ NCUST(2), MAXLN(2), DELAY(2), XMDELAY(2),
- DELAY(2), DELAYX(2), CLTIME(2), CTIME(2), CTIME(2),
- PROB(2), PROBL(2), LTOT(2), STOT(2)
EQUIVALENCE (IBUFF(1), CLOCK(1)), (IBUFF(201), KIM(1)),
- (IBUFF(301), LQ(1))

HISTORY ENTRY

CLOCK(I13)=TIME-PTIME
KIM(I13)=K1+4
I13=I13+1: NTOT=NTOT+1
IF(I13.NE.101) GO TO 4
CALL BUFFEROUT(7,1,IBUFF,400,IND)
1 IF(IND.EQ.1) GO TO 1
1 GO TO (1,2,2)
2 OUTPUT(101): BUFFERING ERROR 4
3 I13=1

SERVER BUSY

4 IBUSY(K1)=1
11=I+2

SELECT QUEUE DISCIPLINE
GO TO (10,20,30,40) IDSPLN(I)
FIFO

```



```

10 STIME=TIM(1,II)
K=1
GO TO 50
C
C
20 STIME=TIM(KP(I),II)
WAIT=TIME-TIM(KP(I),I)
K=KP(I)
GO TO 50
C
C
30 STIME=TIM(1,II)
K=1
DO 35 J=2*KP(I)
35 IF(TIM(J,II)*LT.*STIME) STIME=TIM(J,II); K=J
WAIT=TIME-TIM(K,I)
GO TO 50
C
C
RANDOM
C
40 PIECE=1.0/KP(I)
IX4=IX4-.4099; RN=0.5+IX4*.5960464E-7
K=1
REPEAT 45 WHILE (K*PIECE).LT.RN
45 K=K+1
STIME=TIM(K,II)
WAIT=TIME-TIM(K,I)
C
C
ADJUST ARRAY
C
50 DO 55 J=K*KP(I)
51 TIM(J,I)=TIM(J+1,I)
55 TIM(J,I)=TIM(J+1,I)
KP(I)=KP(I)-1
C
C
MINIMUM SERVICE TIME 1.2 SECONDS (TIME FOR MOTION)
STIME=A MAX(STIME,1.2)
C
C
STATISTICS
C
60 DELAY(I)=DELAY(I)+WAIT
DELAYSQ(I)=DELAYSQ(I)+WAIT**2
IF(WAIT.GT.XMDELAY(I)) XMDELAY(I)=WAIT
IF(WAIT.GT.0.0) DELAYO(I)=DELAYO(I)+1

```



```
IF(WAIT>XD) DELAYX(I)=DELAYX(I)+1
CLTIME(I)=CLTIME(I)+TIME
CLTIMSQ(I)=CLTIMSQ(I)+TIME**2
SET END=OF-SERVICE-TIME
ETIME(K1)=TIME+TIME
LINE(I)=LINE(I)-1
RRETURN
END
C
C
```



```

C          SUBROUTINE END SRV
C          PURPOSE: RECORD END-OFF-SERVICE IN HISTORY AND DETERMINE
C          IF BEGIN SERVICE FOLLOWS
C
C          SUBROUTINE END SRV
C          DIMENSION NO(6)
C          DATA NO/6*0/
C          COMMON /SET6/ CLOCK(100),KIM(100),LOG(100)
C          COMMON /SET7/ ETIME(9),I13,K1,TIM(115*4)
C          COMMON /SET9/ IDIST(4),RATE(4),KK(4),CV(4),NFDBK,IDSPLN(2),
C          -IDUAL,IBUSY(6),KP(2),LINE(2),TIME,PTIME,PTIME,RTIME,IX1,
C          -IX2,IX3,IX4,XD,LQ,MAXQ,P
C          COMMON /SET10/ NTOT,ITYPE
C          COMMON /SET11/ NCUST(2),MAXLN(2),DELAY(2),XMDELAY(2),
C          -DELAYO(2),DELAYX(2),CLTIME(2),CLTMSQ(2),CTIMSQ(2),
C          -PROBO(2),PROBL(2),LTOT(2),ISTOT(2)
C          EQUIVALENCE (IBUFF(1),CLOCK(1)),(IBUFF(201),KIM(1)),
C          -(IBUFF(301),LOG(1))

C          SERVER NOT BUSY
C          IBUSY(K1)=0
C          IF(ITYPE.NE.4) GO TO 10
C          DETERMINE IF FEEDBACK
C          IX3=IX3*4099: RN=0.5+IX3*: 5960464E-7
C          IF(RN.GT.P) IFDBK=1: NFDBK=NFDBK+1
C          HISTORY ENTRY
C          10 CLOCK(I13)=TIME-PTIME
C          ALTERNATE IMAGES
C          IF(NO(K1).EQ.0) NO(K1)=1: GO TO 100
C          NO(K1)=0
C          100 KIM(I13)=K1+10+6*: NO(K1)
C          IF(IFDBK.EQ.1) LCG(I13)=2: GO TO 11
C          IF(ITYPE.EQ.3) LOG(I13)=2: GO TO 11

```



```

11 LOG(I3)=1
12 I3=I3+1: NTOT=NTOT+1
13 IF(I3.NE.101) GO TO 15
14 CALL BUFFEROUT(7,1,IBUFF,400,IND)
15 IF(IND.EQ.1) GO TO 12
16 GO TO (12,14,13,13) IND
17 OUTPUT(101):BUFFERING ERROR 3
18 I3=1

C TIME STATISTICS
C 15 CALL ACCUM
C IN CASE OF BEGIN-SERVICE
C LOG(I3)=6
C IF(IFDBBK.EQ.0) GO TO 17
C CAUSES DELAY
C IF(LINE(1).EQ.0) LOG(I3)=7
C CALL TIMES: KP(I)=KP(I)+1: CALL BEGSRV: KP(I)=KP(I)-1:
C -CALL ARRIV: RETURN
C QUEUE EMPTY?
C 17 IF(LINE(I).EQ.0) ETIME(K1)=99999.: GO TO 20
C CALL BEGSRV
C IF(ITYPE.NE.3) RETURN
C FEEDBACK SERVICE TIME
C I=2: CALL TIMES
C IF(KP(2).NE.1) GO TO 19
C FEEDBACK TO EMPTY QUEUE?
C
C TIM(KP(1)+1,1)=TIM(KP(2),2)
C TIM(KP(1)+1,3)=TIM(KP(2),4)
C KP(2)=KP(2)-1
C ETIME(7)=TIM(KP(1)+1,1)
C
19 RETURN
END

```



卷之三

SUBDIVISION ACCORDING

PURPOSE: GATHER TIME STATISTICS AND ADVANCE STATE CHANGE VARIABLES

```

SUBROUTINE ACCUM
COMMON /SET7/ ETIME(9), I•13, K1, TIM(115•4)
COMMON /SET9/ IDIST(4), RATE(4), KK(4), CV(4), NFDBK, IFDBK, IDSPLN(2),
COMMON /IBUSY/ (6), KP(2), LINE(2), TIME, PTIME(2), RUNTIME, IX1,
COMMON /IBUSY/ (6), KP(2), LINE(2), TIME, PTIME(2), RUNTIME, IX1,
COMMON /IX4/ IX4•XD, LQ•MAXQ, P
COMMON /SET10/ NTOT, ITYPE
COMMON /SET11/ NCUST(2), MAXLN(2), DELAYSSQ(2), XMDELAY(2),
COMMON /SET12/ NCUST(2), MAXLN(2), CLTIME(2), CLTMSQ(2), CUSTIM(2),
COMMON /DELAY0/ (2), DELAYX(2), CLTIME(2), CLTMSQ(2), CUSTIM(2),
COMMON /PROBL/ (2), PROBL(2), LTOT(2), ISTOT(2),
COMMON /PROBO/ (2), PROBO(2), LTOT(2), ISTOT(2),
Z=TIME-PTIME(I)
CUSTIM(I)=CUSTIM(I)+LINE(I)**Z
CTIMSQ(I)=CTIMSQ(I)+(LINE(I),Z)**2
IF(LINE(I).GT.0) PROBO(I)=PROBO(I)+Z
IF(LINE(I).GT.LQ) PROBL(I)=PROBL(I)+Z
ADVANCE STATE CHANGE VARIABLES TO NOW
PTIME(I)=TIME
PTIME=TIME
RETURN
END

```

۲۲۵



SUBROUTINE TIMES

PURPOSE: TO GENERATE AND STORE ARRIVAL (OR IDLE) TIMES  
AND SERVICE TIMES

```
COMMON /SET7/ ETIME(9),I13,K1,TIM(115,4)
COMMON /SET9/ IDIST(4),RATE(4),KK(4),CV(4),NFDBK,IFDBK,IDSPLN(2),
- IDUAL,IBUSY(6),KP(2),LINE(2),TIME,PTIME,PTIME,RUNTIME,IX1,
- IX2,IX3,IX4,XD,LQ,MAXQ,P
COMMON /SET10/ NTOT,ITYPE
COMMON /SET11/ NCUST(2),MAXLN(2),DELAY(2),DELAYSQ(2),XMSQ(2),
- DELAYO(2),DELAYX(2),CLTIME(2),CLTMSQ(2),CUSTIM(2),CTIMSQ(2),
- PROBL(2),PROBL(2),LTOT(2),ISTOT(2)

FIRST TIME THROUGH GETS ARRIVAL (OR IDLE) TIME
IDLE TIMES STORED IN COLUMN 2
I2=1

FEEDBACK? SLIP INTO QUEUE TIME ARRAY
IF(IFDBK.EQ.1) TIM(KP(1)+2,I2)=TIM(KP(1)+1,I2);
-TIM(KP(1)+2,I2+2)=TIM(KP(1)+1,I2+2);
-TIM(KP(1)+1,I2)=TIME; I2=I+2; GO TO 1
IF(ITYPE.NE.5) GO TO 1

DUAL ARRIVAL STREAMS?
IF(IDUAL.EQ.1.AND.I.EQ.2) TIM(KP(2)+1,2)=TIM(KP(1)+1,1); I2=I+2

SELECT DISTRIBUTION
1 GO TO 10,20,30) IDIST(I2)

K=ERLANG
10 U=1.0/(KK(I2)*RATE(I2))
SV=0.0
DO 11 J=1,KK(I2)
11 GENERATE RANDOM NUMBER BETWEEN 0 AND 1
```



```

C      IX1=IX1*4099;  RN=0.5+IX1*.5960464E-7
11     SV=SV-U*ALOG(RN)
      GO TO 88
C      HYPEREXPONENTIAL
20     IX1=IX1*4099;  RN1=0.5+IX1*.5960464E-7
      IX2=IX2*4099;  RN2=0.5+IX2*.5960464E-7
      U=1.0/RATE(I2)
      P=1.0/(CV(I2)*P2)
      UX=0.2928932
      IF(P*GE.RN1) UX=((U-((1.0-P)*UX))/P
      SV=-UX*ALOG(RN2)
      GO TO 88

      DEGENERATE
30     SV=1.0/RATE(I2)
88     IF(I2.NE.1) GO TO 89
      MINIMUM INTER-ARRIVAL TIME=0.5 ( MOTION TIME )
      -GO TO 87
      IF(I TYPE.EQ.2 .OR. I TYPE.EQ.5) SV=AMAX(SV,0.5);
      MINIMUM INTER-ARRIVAL TIME=0.3 ( MOTION TIME )

      SV=AMAX(SV,0.3)
87     TIM(KP(I)+1,I2)=SV+TIME
      I2=I+2
      GO TO 1
89     TIM(KP(I)+1,I2)=SV
      IF(I TYPE.NE.3) GO TO 95
      IF(KP(2).EQ.0) GO TO 91
      NEXT END-IDLE TIME AT TAIL OF ARRAY
      TEMP1=TIM(KP(I)+1,I2)
      TEMP2=TIM(KP(I)+1,4)
      JEKP(I)
      REPEAT 90 WHILE TEMP1.GT.TIM(J,2):TIM(J+1,4)=TIM(J,4) J.GT.0
90     J=J-1
      TIM(J+1,I2)=TEMP1: TIM(J+1,4)=TEMP2
      KP(2)=KP(2)+1
      RETURN
95     END

```



## SUBROUTINE STATISTICS

PURPOSE: DISPLAY THE STATISTICS PAGE

```

SUBROUTINE STATISTICS
DIMENSION IWORD(11)
REAL LT*LC
COMMON /STAT/ QT(2),BT(2),LT(2),QC(2),BC(2),LC(2),
- DC(2),WC(2),DM(2),DO(2),DX(2),QD(2),QV(2),
- QL(2),DV(2),SV(2),PL,NC(2)
COMMON /SET5/ IDEV,IER
COMMON /SET10/ INTOT,TYPE
COMMON (16X,STATISTICS,18X),Q1,6X,02,4X),Q2,6X,02,4X),
FORMAT(7X,Q1,6X,02,4X),Q3,3X,F6,2,2X,F6,2,2X),
FORMAT(7X,QT,3X,F6,2,2X,F6,2,2X),FORMAT(7X,BT,3X,F6,
- 2,2X,F6,2,2X),FORMAT(7X,LT,3X,F6,2,2X,F6,2,2X),
FORMAT(7X,DC,3X,F6,2,2X,F6,2,2X),FORMAT(7X,SC,3X,F6,
- 2,2X,F6,2,2X),FORMAT(7X,WC,3X,F6,2,2X,F6,2,2X),
FORMAT(7X,DM,3X,F6,2,2X,F6,2,2X),FORMAT(7X,DO,3X,F6,
- 2,2X,F6,2,2X),FORMAT(7X,DV,3X,F6,2,2X,F6,2,2X),
FORMAT(7X,NC,3X,F6,2,2X,F6,2,2X),FORMAT(12X,ENTER,1,CONTINUE,1,
- 4,1,2,PARAMETER LISTING,5X)
14 ENCODE(44,1,IWORD),CALL TEXT0(IDEV,IWORD,11,6,7,2,3,IER)
ENCODE(44,2,IWORD),CALL TEXT0(IDEV,IWORD,11,8,7,2,3,IER)
CALL TEXT0(IDEV,IWORD,QT(1),QT(2),QC(1),QC(2)
- ENCODE(44,3,IWORD,QT(1),QT(2),2,3,IER)
CALL TEXT0(IDEV,IWORD,4,1,6T(2),2,3,IER),BC(2)
CALL TEXT0(IDEV,IWORD,5,1,13*1T(2),2,3,IER),LC(2)
ENCODE(44,5,IWORD,LT(1),LT(2),2,3,IER)
CALL TEXT0(IDEV,IWORD,6,1WORD,DC(1),DC(2),DV(2)
- ENCODE(44,7,IWORD,SC(1),SC(2),SV(1),SV(2)
CALL TEXT0(IDEV,IWORD,8,IWORD,WC(1),WC(2),QV(1),QV(2)
- CALL TEXT0(IDEV,IWORD,9,IWORD,WC(1),WC(2),QV(1),QV(2)

```

CCCCCCCC



```

ENCODE(44, 9, IWORD) DM(1), DM(2), QM(1), QM(2)
CALL TEXT0 (IDEV, IWORD, 11, 25, 7, 2, 3, IER)
ENCODE(44, 10, IWORD) DO(1), DO(2), QO(1), QO(2)
CALL TEXT0 (IDEV, IWORD, 11, 27, 7, 2, 3, IER)
ENCODE(44, 11, IWORD) DX(1), DX(2), QL(1), QL(2)
CALL TEXT0 (IDEV, IWORD, 11, 29, 7, 2, 3, IER)
ENCODE(44, 12, IWORD) NC(1), NC(2), PL
CALL TEXT0 (IDEV, IWORD, 11, 32, 7, 2, 3, IER)
ENCODE(44, 13, IWORD)
CALL TEXT0 (IDEV, IWORD, 11, 34, 7, 2, 3, IER)
ENCODE(44, 14, IWORD)
CALL TEXT0 (IDEV, IWORD, 11, 36, 7, 2, 3, IER)
RETURN
END

```



CCCCCCCCCCCC

SUBROUTINE PARAMETERS

PURPOSE: DISPLAY THE CURRENT PARAMETERS

```
COMMON /SET3/X(14)•Y(25)•MD(6)•IC,IC1•NULL
COMMON /SET5/IDEV•IER
COMMON /SET8/IHNG(10)•ENDD(5,22)•IFLAG(22)•ITOT•IPOS(2),
COMMON /IMAN/IN(2)•IN(2)•NSIM(10)•ITQ(2)•NP0P,ITPOP(2)•XX(5,22),
-XXX(5,22)•YY(5,22)•YYY(5,22)•IMD(5),
COMMON /SET9/IDIST(4)•RATE(4)•KK(4)•CV(4)•NFDBK,IDSPLN(2),
-IMD•IBUSY(6)•KP(2)•LINE(2)•TIME,PSTIME,PTIME,IX1,
-IX2•IX3•IX4•XD•LQ•MAXQP
COMMON /SET10/NTOT•ITYPE
FORMAT(7X•IC•8X•14•6X•IC1•7X•14•3X)
12 FORMAT(10X•1•9X•2•9X•3•9X•4•3X)
3 FORMAT(14•6X•14•6X•14•6X•14•6X•14•3X)
4 FORMAT(•IDIST •14•6X•14•6X•14•6X•14•3X)
5 FORMAT(•RATE •14•6X•14•6X•14•6X•14•3X)
6 FORMAT(•CV •F5•2•5X•F5•2•5X•F5•2•5X,F5•2,3X)
7 FORMAT(•IDSPLN •F5•2•5X•F5•2•5X•F5•2,3X)
8 FORMAT(7X•LQ•8X•14•6X•MAXQ(6X,14•3X)
9 FORMAT(7X•XD•7X•F5•2•6X•NP0P•6X,14•3X)
10 FORMAT(7X•IDUAL •5X•14•6X•P•8X•F5•2,3X)
11 FORMAT(12X•IDUAL •5X•14•6X•P•8X•F5•2,3X)
12 FORMAT(12X•1•CONTINUE•14X)
13 FORMAT(19X•2•STATISTICS•12X)
ENCODE(44,1•IWORD) ITYPE, RUNTIME
CALL TEXT0 (IDEV•IWORD•11•8,7,2,3,IER)
ENCODE(44,2•IWORD) IC,IC1
CALL TEXT0 (IDEV•IWORD•11,10,7,2,3,IER)
ENCODE(44,3•IWORD)
CALL TEXT0 (IDEV•IWORD•11•13•7•2•3•IER)
ENCODE(44,4•IWORD) IDIST(1)•7•2•3•IER
CALL TEXT0 (IDEV•IWORD•11•15•7•2•3•IER)
ENCODE(44,5•IWORD) KK(1)•KK(2)•KK(3)•KK(4)
CALL TEXT0 (IDEV•IWORD•11•17•7•2•3•IER)
ENCODE(44,6•IWORD) RATE(1)•RATE(2)•RATE(3), RATE(4)
CALL TEXT0 (IDEV•IWORD•11•19•7•2•3•IER)
ENCODE(44,7•IWORD) CV(1)•CV(2)•CV(3)•CV(4)
CALL TEXT0 (IDEV•IWORD•11•21•7•2•3•IER)
```



```

ENCODE (44, 8*IWORD) IDSPLN(1)*IDSPLN(2)
CALL TEXTO (IDEV*IWORD,11,23,7,2,3,IER)
ENCODE (44, 9*IWORD) LQ,MAXQ
CALL TEXTO (IDEV*IWORD,11,26,7,2,3,IER)
ENCODE (44, 10*IWORD) XD,NPOP
CALL TEXTO (IDEV*IWORD,11,28,7,2,3,IER)
ENCODE (44, 11*IWORD) IDUALP
CALL TEXTO (IDEV*IWORD,11,30,7,2,3,IER)
ENCODE (44, 12*IWORD)
CALL TEXTO (IDEV*IWORD,11,33,7,2,3,IER)
ENCODE (44, 13*IWORD)
CALL TEXTO (IDEV*IWORD,11,35,7,2,3,IER)
RETURN
END

```



## SUBROUTINE DRIVER

## PURPOSE : TO CAUSE THE DYNAMIC DISPLAY OF THE TIME HISTORY

THE BRITISH LINEAGE: SETI AGS. E. I. AGS. REMAN

```

SUBROUTINE DRIVER
DIMENSION IBUFF(400)
COMMON /SET6/ CLOCK(100), KIM(100), LOG(100)
COMMON /SET8/ CHNG(10), ENDD(5,22), IFLAG(22), ITOT(22), IPOP(22), IPOS(22),
- ILM(2), IMAN(7,22), NSIM(5,22), NQ(22), ITQ(2,22), ITD(5,22),
- XX(5,22), YY(5,22), YY(5,22), NTOT, ITYPE
COMMON /SET10/ NTOT, ITYPE
COMMON /SET12/ IDONE, I3, IM
EQUIVALENCE (IBUFF(1), CLOCK(1)), (IBUFF(201), KIM(1)),
- (IBUFF(301), LOG(1))
10 IDONE=1
CALL BUFFERIN (7,1,IBUFF,400,IND)
11 IF (IND.EQ.1) GO TO 11
111 GO TO (11,13,12,12) IND
12 OUTPUT(101); BUFFERIN ERROR
13 I3=1

DISALLOWS TOO SMALL A VALUE AFTER BUFFERIN
14 IF (CLOCK(13).LE.0.1) CLOCK(13)=.1
SET MAX (HYBRID LIMITATION)
15 IF (CLOCK(13).GT.99.) CLOCK(13)=99.
SET TIMER FOR NEXT EVENT
16 CALL DAC(1,CLOCK(13)/100.)
CALL COMPUTE
CALL ENABLE
CALL WAIT

119 IF (IDONE.EQ.1) GO TO 19
CALL DISABLE
IF (IDONE.EQ.2) GO TO 10

```



30 REWIND 7: RETURN  
END



CCCCC SUBROUTINE SETFLAGS

PURPOSE: SET IMAGE FLAGS AS PER TIME HISTORY AT THE  
OCCURANCE OF A TIMING INTERRUPT (53 INTERRUPT)

```
COMMON /SET6/ CLOCK(100),KIM(100),LOG(100),
COMMON /SET8/ CHNG(10),ENDD(5,22),IFLAG(22),ITOT,IPOS(2),
-ILN(2),IMAN(7,22),NSIM,NQ(22),ITQ(2,2),NPDP(2),XX(5,22),
-XXX(5,22),YY(5,22),IMD(5),
COMMON /SET10/ NTOT,ITYPE
COMMON /SET12/ IDONE,I3,IM
CCCCC SET CORRECT IMAGE FLAG TO MOVEMENT LOGIC VALUE AS PER HISTORY
10 IFLAG(KIM(I3))=LOG(I3)
I3=I3+1
ITOT=ITOT+1
IF(ITOT.GT.NTOT) IDONE=3: RETURN
IF(I3.GT.100) IDONE=2: RETURN
CCCCC IF TIME TO NEXT EVENT IS SMALL, SET THE FLAG
IF(CLOCK(I3).LE..0.1) GO TO 10
SET MAX (HYBRID LIMITATION)
IF(CLOCK(I3).GT..99.) CLOCK(I3)=99.
CCCCC SET THE TIMER FOR THE NEXT EVENT INTERRUPT
CALL DAC(1,CLOCK(I3)/100.)
CALL COMPUTE
RETURN
END
```



```

C SUBROUTINE FLAGS
C
C PURPOSE: DISPLAY AND MOVE THE DYNAMIC IMAGES
C

```

```

SUBROUTINE FLAGS
COMMON /SET5/ IDEV, IER
COMMON /SET8/ CHNG(10), ENDD(5,22), IFLAG(22), ITOT, IPOS(2),
- ITN(2), IMAN(7,22), NSIM, NQ(2), ITQ(2,2), NPOP, ITPOP(2), XX(5,22),
- XXX(5,22), YY(5,22), YYYY(5,22), IMD(5)
COMMON /SET10/ NTOT, ITYPE
COMMON /SET12/ IDONE, I3, IM
1 FORMAT (14)

C CHECK ALL THE FLAGS

IM=22
REPEAT 99 WHILE IM.GT.0
IF(IFLAG(IM).GT.0) GO TO 10, 20, 30, 40, 50, 60, 70,
- 71, 72, 73, 74, 75) IFLAG(IM)
99 IM=IM-1
RETURN

C HORIZONTAL MOVEMENT RIGHT (ARRIVALS, STRAIGHT DEPARTURES)

10 IF(XX(1,IM).NE.XXX(1,IM)) OR.REMAN: IM=10) GO TO 5
IF(IM.GE.17) IM=IM-12: CALL REMAN: IM=IM+12; GO TO 5
IM=IM-6: CALL GRAPHO(IDEV, IMAN(1,IM), 7, NSIM+IM, IER)
5 TEST=XX(1,IM)+CHNG(1)
IF(TEST.LE.ENDD(5,IM)) GO TO 18
IF(IM.GT.4) GO TO 13
12 IF(ITYPE.NE.3) GO TO 15
NPOP=NPOP-1: ENCODE(4,1, ITPOP) NPOP
CALL TEXTO(IDEV, ITPOP, 2, ILN(2), IPOS(2), 3, 3, IER)
NQ(1)=NQ(1)+1
IF(NQ(1).LE.0) GO TO 13
ENCODE(4,1, ITQ(1,1)) NQ(1)
CALL TEXTO(IDEV, ITQ(1,1), 2, ILN(1), IPOS(1), 3, 3, IER)
GO TO 13
15 IF(ITYPE.NE.5) IT=1: GO TO 6
IF(IM.EQ.1 .OR. IM.EQ.3) IT=1: GO TO 6
IT=2

```



```

6  NQ(IT)=NQ(IT)+1: ENCODE(4*1,ITQ(1,IT),NQ(IT));
-IF(NQ(IT)>0) GOTO 17
-CALL TEXT TO (IDEV,ITQ(1,IT),2,ILN(IT),IPOS(IT),3,3,IER)
13 CALL REMAN: GO TO 99
18 DO 19 I=2,6
19 IMAN(I,IM)=XX(J,IM)+CHNG(1)
      YY(J,IM),IMD(J))
      GO TO 99

C C HORIZONTAL MOVEMENT RIGHT (WHEN A COURSE CHANGE WILL O
C C
C C
20 IF(XX(1,IM).NE.XXX(1,IM) OR IM.LE.10) GO TO 7
  IF(IM.GE.17) IM=IM-12: CALL REMAN: IM=IM+12: GO TO 7
  IM=IM-6: CALL REMAN: IM=IM+6
  7 CALL GRAPHO(I,DEV,IMAN(1,IM),7,NSIM+IM,IER)
  TEST=XX(1,IM)+CHNG(3)
  IF(TEST.GE.ENDD(1,IM)) IFLAG(IM)=3: GO TO 99
  DO 28 I=2,6
28 IMAN(J,IM)=XX(J,IM)+CHNG(3)
      YY(J,IM),IMD(J))
      GO TO 99

C C VERTICAL MOVEMENT DOWNWARD
C C
C C
30 CALL GRAPHO(I,DEV,IMAN(1,IM),7,NSIM+IM,IER)
  TEST=YY(1,IM)-CHNG(3)
  IF(TEST.GE.ENDD(2,IM)) GO TO 31
  IF(IM.LE.4) CALL REMAN: GO TO 99
  IFLAG(IM)=4: GO TO 99
  DO 38 I=2,6
38 IMAN(J,IM)=YY(J,IM)-CHNG(3)
      YY(J,IM),IMD(J))
      GO TO 99

C C HORIZONTAL MOVEMENT TO THE LEFT
C C
C C
40 CALL GRAPHO(I,DEV,IMAN(1,IM),7,NSIM+IM,IER)
  TEST=XX(1,IM)-CHNG(3): 2*0
  IF(TEST.GE.ENDD(3,IM)) GO TO 47
  IFLAG(IM)=5: GO TO 99
  45 DO 48 I=2,6
47 J=I-1
  XX(J,IM)=XX(J,IM)-CHNG(3): 2*0
  IMAN(J,IM)=IPACK(XX(J,IM),YY(J,IM),IMD(J))
  GO TO 99
48 IMAN(I,IM)=IPACK(XX(J,IM),YY(J,IM),IMD(J))
      GO TO 99

```



C C VERTICAL MOVEMENT UPWARD

50 CALL GRAPHO (IDEV,IMAN(1,IM),7,NSIM+IM,IER)  
TEST=YY(1,IM)+CHNG(4)  
IF(TEST.LE.ENDED(4,IM)) GO TO 57  
IF(ITYPE.NE.4) GO TO 55  
NQ(1)=NQ(1)+1  
IF(NQ(1).LE.0) GO TO 56  
FNCODE(4,1,ITQ(1,1)) NQ(1)  
CALL TEXTO (IDEV,ITQ(1,1),2,ILN(1),IPOS(1),3,3,IER)  
GO TO 56  
55 NPOP=NPOP+1; ENCODE(4,1,ITPOP) NPOP;  
-CALL TEXTO (IDEV,ITPOP,2,ILN(2),IPOS(2),3,3,IER)  
56 CALL REMAN; GO TO 99  
57 DO 58 I=2,6  
J=I-1  
YY(J,IM)=YY(J,IM)+CHNG(4)  
58 IMAN(1,IM)=IPACK(XX(J,IM),YY(J,IM),IMD(J))  
GO TO 99

C C MOVEMENT TO-SERVICE

60 IF(XX(1,IM).NE.XXX(1,IM)) GO TO 61  
IF(ITYPE.NE.5.OR.ILT.8) GO TO 65  
NQ(2)=NQ(2)-1  
IF(NQ(2).GE.0) ENCODE(4,1,ITQ(1,2)) NQ(2);  
-CALL TEXTO (IDEV,ITQ(1,2),2,ILN(2),IPOS(2),3,3,IER)  
GO TO 61  
65 NQ(1)=NQ(1)-1  
IF(NQ(1).LT.0) GO TO 61  
FNCODE(4,1,ITQ(1,1)) NQ(1)  
CALL TEXTO (IDEV,ITQ(1,1),2,ILN(1),IPOS(1),3,3,IER)  
61 CALL GRAPHO (IDEV,IMAN(1,IM),7,NSIM+IM,IER)  
TEST=XX(1,IM)+CHNG(2)  
IF(TEST.GT.ENDD(1,IM)) IFLAG(IM)=0; GO TO 99  
DO 68 I=2,6  
J=I-1  
XX(J,IM)=XX(J,IM)+CHNG(2)  
YY(J,IM)=YY(J,IM)+CHNG(IM)  
68 IMAN(1,IM)=IPACK(XX(J,IM),YY(J,IM),IMD(J))  
GO TO 99

C C CAUSE DELAY

70 IFLAG(IM)=8 : GO TO 99  
71 IFLAG(IM)=9 : GO TO 99  
72 IFLAG(IM)=10: GO TO 99



```
73 IFLAG(1M)=11:: GO TO 99
74 IFLAG(1M)=12:: GO TO 99
75 IFLAG(1M)=6 :: GO TO 99
END
```



# SUBROUTINE REMAN PURPOSE: BLANK OUT IMAGES, AND REPOSITION

```

SUBROUTINE REMAN 10
COMMON /SET5/ IDEV, IER
COMMON /SET8/ CHNG(10), ENDD(5,22), IFLAG(22), ITOT, IPOS(2),
- ILN(2), IMAN(7,22), NSIM, NQ(2), ITQ(2,2), NPOP, ITPOP(2),
- XXX(5,22), YY(5,22), IMD(5),
COMMON /SET10/ NTOT, ITYPE
COMMON /SET12/ IDONE, IM
IF (IM.LE.4 .OR. IM.GE.11) IFLAG(IM)=0
DO 10 K=2,6
10 IMAN(K,IM)=0
CALL GRAPHO (IDEV, IMAN(1,IM), 7, NSIM+IM, IER)
DO 20 K=2,6
20 IMAN(K,IM)=IPACK(XX(J,IM), YY(J,IM))
J=K-1
XX(J,IM)=XXX(J,IM)
YY(J,IM)=YYY(J,IM)
20 IMAN(K,IM)=IPACK(XX(J,IM), YY(J,IM), IMD(J))
RETURN
END

```



SUBROUTINE GGC

PURPOSE: DISPLAY STATIC PICTURE FOR G/G/C MODEL

COMMON /SET1/ISERV1(37),IPATH1(19),IPATH2(19),IPATH3(19),  
-IPATH7(8),  
-COMMON /SET2/ISERV2(19),IQUE1(7),IQUE2(12),IQUE3(7),IPATH4(10),  
-IPATH5(10),IPATH9(7),IPATH10(5),IPATH11(4),  
-IPATH12(4),IPATH13(6),  
-COMMON /SET3/X(14),Y(25),MD(6),IC,IC1,NULL  
COMMON /SET5/IDEV,IER  
DO 10 I=1,IC

10 MD(I)=1  
CALL SERV1  
CALL GRAPHO (IDEV,ISERV1,6\*IC+1,1,IER)  
CALL PATH1  
CALL GRAPHO (IDEV,IPATH1,3\*IC+1,2,IER)  
CALL PATH2  
CALL GRAPHO (IDEV,IPATH2,3\*IC+1,3,IER)  
CALL GRAPHO (IDEV,IPATH12,4\*IC+1,4,IER)  
CALL GRAPHO (IDEV,IQUE1,7\*IC+1,5,IER)  
RETURN  
END

CCCCCCCC



**SUBROUTINE LOSSES**

**PURPOSE: DISPLAY STATIC PICTURE FOR G/G/C WITH LOSSES MODEL**

```

SUBROUTINE LOSSES
COMMON /SET1/ISERV1(37),IPATH1(19),IPATH2(19),IPATH3(19),
-IPATH7(8)
COMMON /SET2/ISERV2(19),IQUEUE1(7),IQUEUE2(12),IQUEUE3(7),IPATH4(10),
-IPATH5(10),IPATH9(7),IPATH10(5),IPATH11(4),
-IPATH12(4),IPATH13(6)
COMMON /SET3/X(14),Y(25),MD(6),IC,IC1,NULL
COMMON /SET5/IDEV,IER
CALL GGC
CALL GRAPHO (IDEV,IPATH1,4,6,IER)
RETURN
END

```



## SUBROUTINE FEEDBACK PURPOSE: DISPLAY STATIC PICTURE FOR G/G/C WITH FEEDBACK

```

SUBROUTINE FEEDBK
COMMON /SET1/ISERV1(37),IPATH1(19),IPATH2(19),IPATH3(19),
-IPATH7(8)
COMMON /SET2/ISERV2(19),IQUE1(7),IQUE2(12),IPATH4(10),
-IPATH5(10),IPATH9(7),IPATH10(5),IPATH11(4),
-IPATH12(4),IPATH13(6)
COMMON /SET3/X(14),Y(25),MD(6),IC,ICI,NULL
COMMON /SET5/IDEV,IER
CALL GGC
DO 10 I=1,4
MD(1)=1
IF(IC.LT.6)MD(1)=0:IF(IC.LT.4)MD(2)=0:IF(IC.LT.2)MD(3)=0
CALL PATH7
CALL GRAPHO (IDEV,IPATH7,8,6,IER)
CALL GRAPHO (IDEV,IPATH10,5,7,IER)
RETURN
10

```



ੴ ਸਤਿਗੁਰ

## SUBROUTINE FINITE

PURPOSE: DISPLAY STATIC PICTURE FOR G/G/C WITH FINITE SOURCE

```

SUBROUTINE FINITE
COMMON /SET1/ISERV1(37),IPATH1(19),IPATH2(19),IPATH3(19),
-IPATH7(8)
COMMON /SET2/ISERV2(19),IQUE1(7),IQUE2(12),IQUEUE3(7),IPATH4(10),
-IPATH5(10),IPATH9(7),IPATH10(5),IPATH11(4),
-IPATH12(4),IPATH13(6)
COMMON /SET3/X(14),Y(25),MD(6),IC,IC1,NULL
COMMON /SET5/IDEV,IER
DO 10 I=1,4
10 MD(I)=1
IF(IC.LT.6)MD(1)=0;IF(IC.LT.4)MD(2)=0;IF(IC.LT.2)MD(3)=0
CALL PATH7
DO 11 I=1,IC
11 MD(I)=1
CALL SERV1
CALL PATH1
CALL PATH3
CALL GRAPHO ((IDEV,ISERV1,6,IC+1,1,IER)
CALL GRAPHO ((IDEV,IQUE1,7,2,IER)
CALL GRAPHO ((IDEV,IQUE3,7,3,IER)
CALL GRAPHO ((IDEV,IPATH1,3,IC+1,4,IER)
CALL GRAPHO ((IDEV,IPATH3,3,IC+1,5,IER)
CALL GRAPHO ((IDEV,IPATH7,8,6,IER)
CALL GRAPHO ((IDEV,IPATH9,7,7,IER)
RETURN
END

```



CC  
CC  
CC  
CC  
CC  
CC  
CC  
CC  
CC  
CC

SUBROUTINE TWOGCC  
PURPOSE: DISPLAY STATIC PICTURE FOR 2 G/G/C MODEL

COMMON /SET1/ISERV1(37),IPATH1(19),IPATH2(19),IPATH3(19),  
IPATH7(8),  
COMMON /SET2/ISERV2(19),IQUE1(7),IQUE2(12),IQUE3(7),IPATH4(10),  
IPATH5(10),IPATH9(7),IPATH10(5),IPATH11(4),  
IPATH12(4),IPATH13(6),  
COMMON /SET3/X(14),Y(25),MD(6),IC,IC1,NULL  
COMMON /SET5/IDEV,IER  
DO 10 I=1,6

10 MD(I)=1  
IF(IC.LT.3) MD(6)=0; IF(IC.LT.2) MD(2)=0  
IF(IC1.LT.3) MD(1)=0; IF(IC1.LT.2) MD(5)=0  
CALL PATH2  
CALL ISERV1  
CALL PATH2  
CALL GRAPHO(IDEV,IQUE2,12,1,IER)  
CALL GRAPHO(IDEV,ISERV1,3,2,IER)  
CALL GRAPHO(IDEV,IPATH1,3,6,IER)  
CALL GRAPHO(IDEV,IPATH4,3,4,IER)  
CALL GRAPHO(IDEV,IPATH5,3,5,IER)  
CALL GRAPHO(IDEV,IPATH2,19,6,IER)  
RETURN  
END



## SUBROUTINE SERV1

PURPOSE: PACK THE SERVICE BOXES

COMMON /SET1/ISERV1(37),IPATH1(19),IPATH2(19),IPATH3(19),  
-IPATH7(8)

```
COMMON /SET3/X(14),Y(25),MD(6),IC,IC1,NULL
ISERV1(1)=IHEAD(0,10)
ISERV1(2)=IPACK(X(11),Y(13),0)
ISERV1(3)=IPACK(X(11),Y(17),MD(1))
ISERV1(4)=IPACK(X(12),Y(17),MD(1))
ISERV1(5)=IPACK(X(12),Y(13),MD(1))
ISERV1(6)=IPACK(X(11),Y(13),MD(1))
ISERV1(7)=0
ISERV1(8)=IPACK(X(11),Y(11),0)
ISERV1(9)=IPACK(X(11),Y(7),MD(2))
ISERV1(10)=IPACK(X(12),Y(7),MD(2))
ISERV1(11)=IPACK(X(12),Y(11),MD(2))
ISERV1(12)=IPACK(X(11),Y(11),MD(2))
ISFRV1(13)=0
ISERV1(14)=IPACK(X(11),Y(18),0)
ISERV1(15)=IPACK(X(11),Y(20),MD(3))
ISERV1(16)=IPACK(X(12),Y(20),MD(3))
ISERV1(17)=IPACK(X(12),Y(18),MD(3))
ISERV1(18)=IPACK(X(11),Y(18),MD(3))
ISERV1(19)=0
ISERV1(20)=IPACK(X(11),Y(6),0)
ISERV1(21)=IPACK(X(11),Y(4),MD(4))
ISERV1(22)=IPACK(X(12),Y(4),MD(4))
ISERV1(23)=IPACK(X(12),Y(6),MD(4))
ISERV1(24)=IPACK(X(11),Y(6),MD(4))
ISERV1(25)=0
ISERV1(26)=IPACK(X(11),Y(21),0)
ISERV1(27)=IPACK(X(11),Y(23),MD(5))
ISERV1(28)=IPACK(X(12),Y(23),MD(5))
ISERV1(29)=IPACK(X(12),Y(21),MD(5))
ISFRV1(30)=IPACK(X(11),Y(21),MD(5))
ISERV1(31)=0
ISERV1(32)=IPACK(X(11),Y(3),0)
ISERV1(33)=IPACK(X(12),Y(1),MD(6))
ISERV1(34)=IPACK(X(12),Y(1),MD(6))
```



```
ISERV1(35)=1PACK(X(12),Y(3),MD(6))
ISERV1(36)=1PACK(X(11),Y(3),MD(6))
ISERV1(37)=0
RETURN
END
```



## SUBROUTINE PATHI

## PURPOSE: PACK THE TO-SERVICE PATHS

故於好尚之物，皆以爲好，故於好尚之物，皆以爲好。

```

SUBROUTINE PATH1
COMMON /SET1/ISERV1(37),IPATH1(19),IPATH2(19),IPATH3(19),
-IPATH7(8)
COMMON /SET3/X(14),Y(25),MD(6),IC,IC1,NULL
IPATH1(1)=IHEAD(1,8)
IPATH1(2)=IPACK(X(10),Y(12),0)
IPATH1(3)=IPACK(X(11),Y(15),MD(1))
IPATH1(4)=0
IPATH1(5)=IPACK(X(10),Y(12),0)
IPATH1(6)=IPACK(X(11),Y(9),MD(2))
IPATH1(7)=0
IPATH1(8)=IPACK(X(10),Y(12),0)
IPATH1(9)=IPACK(X(11),Y(19),MD(3))
IPATH1(10)=0
IPATH1(11)=IPACK(X(10),Y(12),0)
IPATH1(12)=IPACK(X(11),Y(5),MD(4))
IPATH1(13)=0
IPATH1(14)=IPACK(X(10),Y(12),0)
IPATH1(15)=IPACK(X(11),Y(22),MD(5))
IPATH1(16)=0
IPATH1(17)=IPACK(X(10),Y(12),0)
IPATH1(18)=IPACK(X(11),Y(2),MD(6))
IPATH1(19)=0
RETURN
END

```

ੴ ਪੰਡਿੰਦ



CCCCCCCC

## SUBROUTINE PATH2

PURPOSE: PACK THE HORIZONTAL FROM-SERVICE PATHS

ROUTINE: PATH2

```
COMMON /SET1/ ISERV1(37),IPATH1(19),IPATH2(19),IPATH3(19),  
-IPATH7(8)  
COMMON /SET3/X(14),Y(25),MD(6),IC,IC1,NULL  
IPATH2(1)=IHEAD(1,8)  
IPATH2(2)=IPACK((X(12),Y(15),0)  
IPATH2(3)=IPACK((X(14),Y(15),MD(1))  
IPATH2(4)=0  
IPATH2(5)=IPACK((X(14),Y(9),0)  
IPATH2(6)=IPACK((X(12),Y(9),MD(2))  
IPATH2(7)=0  
IPATH2(8)=IPACK((X(12),Y(19),0)  
IPATH2(9)=IPACK((X(14),Y(19),MD(3))  
IPATH2(10)=0  
IPATH2(11)=IPACK((X(14),Y(5),0)  
IPATH2(12)=IPACK((X(12),Y(5),MD(4))  
IPATH2(13)=0  
IPATH2(14)=IPACK((X(12),Y(22),0)  
IPATH2(15)=IPACK((X(14),Y(22),MD(5))  
IPATH2(16)=0  
IPATH2(17)=IPACK((X(14),Y(2),0)  
IPATH2(18)=IPACK((X(12),Y(2),MD(6))  
IPATH2(19)=0  
RETURN  
END
```



CCCCCCCC

## SUBROUTINE PATH3

PURPOSE: PACK THE SHORT HORIZONTAL FROM-SERVICE PATHS

COMMON /SET1/ISERV1(37),IPATH1(19),IPATH2(19),IPATH3(19),  
-IPATH7(8)

```
COMMON /SET3/X(14)*Y(25),MD(6),IC,IC1,NULL
IPATH3(1)=IHEAD(1,8)
IPATH3(2)=IPACK((X(12)*Y(15)*0)
IPATH3(3)=IPACK((X(13)*Y(15)*MD(1))
IPATH3(4)=0
IPATH3(5)=IPACK((X(13)*Y(9)*0)
IPATH3(6)=IPACK((X(12)*Y(9)*MD(2))
IPATH3(7)=0
IPATH3(8)=IPACK((X(12)*Y(19)*0)
IPATH3(9)=IPACK((X(13)*Y(19)*MD(3))
IPATH3(10)=0
IPATH3(11)=IPACK((X(13)*Y(5)*0)
IPATH3(12)=IPACK((X(12)*Y(15)*MD(4))
IPATH3(13)=0
IPATH3(14)=IPACK((X(12)*Y(22)*0)
IPATH3(15)=IPACK((X(13)*Y(22)*MD(5))
IPATH3(16)=0
IPATH3(17)=IPACK((X(13)*Y(2)*0)
IPATH3(18)=IPACK((X(12)*Y(2)*MD(6))
IPATH3(19)=0
RETURN
END
```



CCCCCCCC

## SUBROUTINE PATH7

PURPOSE: PACK A VERTICAL PATH

COMMON /SET1/ISERV1(37),IPATH1(19),IPATH2(19),IPATH3(19),  
-IPATH7(8)

COMMON /SET3/X(14),Y(25),MD(6),IC,IC1,NULL  
-IPATH7(1)=IHEAD(1,8)  
IPATH7(2)=IPACK(X(13),Y(2),0)  
IPATH7(3)=IPACK(X(13),Y(5),MD(1))  
IPATH7(4)=IPACK(X(13),Y(9),MD(2))  
IPATH7(5)=IPACK(X(13),Y(15),MD(3))  
IPATH7(6)=IPACK(X(13),Y(19),1)  
IPATH7(7)=IPACK(X(13),Y(24),MD(4))  
IPATH7(8)=0

RETURN  
END



CCCCCCCC

## SUBROUTINE PACK

PURPOSE: PACK OTHER BOXES AND PATHS

```
COMMON /SET1/ISERV1(37),IPATH1(19),IPATH2(19),IPATH3(19),
-IPATH7(8)
COMMON /SET2/I SERV2(19),IQUE1(7),IQUE2(12),IPATH1(5),IPATH1(4),
-IPATH5(10),IPATH9(7),IPATH10(5),IPATH4(10),
-IPATH12(4),IPATH13(6)
COMMON /SET3/X(14),Y(25),MD(6),IC,IC1,NULL
IQUE1(1)=IHEAD(0,10)
IQUE1(2)=IPACK(X(9),Y(10),0)
IQUE1(3)=IPACK(X(9),Y(14),1)
IQUE1(4)=IPACK(X(10),Y(14),1)
IQUE1(5)=IPACK(X(10),Y(10),1)
IQUE1(6)=IPACK(X(9),Y(10),1)
IQUE1(7)=0
IQUE2(1)=IHEAD(0,10)
IQUE2(2)=IPACK(X(9),Y(4),0)
IQUE2(3)=IPACK(X(9),Y(6),1)
IQUE2(4)=IPACK(X(10),Y(6),1)
IQUE2(5)=IPACK(X(10),Y(4),1)
IQUE2(6)=IPACK(X(9),Y(4),1)
IQUE2(7)=IPACK(X(9),Y(18),0)
IQUE2(8)=IPACK(X(9),Y(20),1)
IQUE2(9)=IPACK(X(10),Y(20),1)
IQUE2(10)=IPACK(X(10),Y(18),1)
IQUE2(11)=IPACK(X(9),Y(18),1)
IQUE2(12)=0
IQUE3(1)=IHEAD(0,10)
IQUE3(2)=IPACK(X(7),Y(8),0)
IQUE3(3)=IPACK(X(3),Y(8),1)
IQUE3(4)=IPACK(X(3),Y(16),1)
IQUE3(5)=IPACK(X(7),Y(16),1)
IQUE3(6)=IPACK(X(7),Y(8),1)
IQUE3(7)=0
ISERV2(1)=IHEAD(0,10)
ISERV2(2)=IPACK(X(6),Y(18),0)
ISERV2(3)=IPACK(X(4),Y(18),1)
ISERV2(4)=IPACK(X(4),Y(20),1)
ISERV2(5)=IPACK(X(6),Y(20),1)
```



```

|||SERV2(6)=IPACK((X(6),Y(18),1)
|||SERV2(7)=0
|||SERV2(8)=IPACK((X(6),Y(17),0)
|||SERV2(9)=IPACK((X(6),Y(13),1)
|||SERV2(10)=IPACK((X(4),Y(13),1)
|||SERV2(11)=IPACK((X(4),Y(17),1)
|||SERV2(12)=IPACK((X(6),Y(17),1)
|||SERV2(13)=0
|||SERV2(14)=IPACK((X(6),Y(21),0)
|||SERV2(15)=IPACK((X(6),Y(23),1)
|||SERV2(16)=IPACK((X(4),Y(23),1)
|||SERV2(17)=IPACK((X(4),Y(21),1)
|||SERV2(18)=IPACK((X(6),Y(21),1)
|||SERV2(19)=0
|||PATH4(1)=IHEAD(1,8)
|||PATH4(2)=IPACK((X(10),Y(5),0)
|||PATH4(3)=IPACK((X(11),Y(5),1)
|||PATH4(4)=0
|||PATH4(5)=IPACK((X(10),Y(5),0)
|||PATH4(6)=IPACK((X(11),Y(9),1)
|||PATH4(7)=0
|||PATH4(8)=IPACK((X(10),Y(5),0)
|||PATH4(9)=IPACK((X(11),Y(2),1)
|||PATH4(10)=0
|||PATH5(1)=IHEAD(1,8)
|||PATH5(2)=IPACK((X(10),Y(19),0)
|||PATH5(3)=IPACK((X(11),Y(19),1)
|||PATH5(4)=0
|||PATH5(5)=IPACK((X(10),Y(19),0)
|||PATH5(6)=IPACK((X(11),Y(22),1)
|||PATH5(7)=0
|||PATH5(8)=IPACK((X(10),Y(19),0)
|||PATH5(9)=IPACK((X(11),Y(15),1)
|||PATH5(10)=0
|||PATH9(1)=IHEAD(1,8)
|||PATH9(2)=IPACK((X(13),Y(24),0)
|||PATH9(3)=IPACK((X(5),Y(24),1)
|||PATH9(4)=IPACK((X(5),Y(16),1)
|||PATH9(5)=IPACK((X(7),Y(12),0)
|||PATH9(6)=IPACK((X(9),Y(12),1)
|||PATH9(7)=0
|||PATH10(1)=IHEAD(1,8)
|||PATH10(2)=IPACK((X(13),Y(24),0)
|||PATH10(3)=IPACK((X(8),Y(24),1)
|||PATH10(4)=IPACK((X(8),Y(12),1)
|||PATH10(5)=0
|||PATH11(1)=IHEAD(1,8)
|||PATH11(2)=IPACK((X(8),Y(12),0)

```



```
IPATH11(3)=IPACK(X(8),Y(25),1)
IPATH11(4)=0
IPATH12(1)=1HEAD(1,8)
IPATH12(2)=1PACK(X(9),Y(12),0)
IPATH12(3)=1PACK(X(1),Y(12),1)
IPATH12(4)=0
IPATH13(1)=1HEAD(1,8)
IPATH13(2)=1PACK(X(9),Y(5),0)
IPATH13(3)=1PACK(X(1),Y(5),1)
IPATH13(4)=1PACK(X(1),Y(19),0)
IPATH13(5)=1PACK(X(9),Y(19),1)
IPATH13(6)=0
RETURN
END
```



```
*****
* SUBROUTINE XES
* PURPOSE: POSITION ALL DYNAMIC IMAGES
*****
C C C C C C C
```

```

SUBROUTINE XES
COMMON /SET3/X(14),Y(25),MD(6),IC,IC1,NULL
COMMON /SET5/IDEV,IER
COMMON /SET8/ICHNG(10),ENDD(5*22),IFLAG(22),ITOT,IPOS(2),
-IQN(2),IMAN(7*22),NSIM,NQ(2),ITQ(2),NPQP,ITPOP(2),XX(5,22),
-XXX(5*22)*YY(5*22)*YYY(5*22),IMD(5)
COMMON /SETX/ A1,A2,A3,A4,61,62,63,S1,S2,S3,S4,S5,S6,S7,S8
DO 10 I=1*22
IMAN(1*I)=I
10 IMAN(1*I)=IFEAD(0,10)
DO 9 J=2*7
IMAN(J)=0
9 CALL GRAPHO(IDEV,IMAN(1*I),7,NSIM+1,IER)
10 XXX(1*01)=XX(1*01)=X(1*01)=X(A1)+.02
YYY(1*01)=YY(1*01)=Y(1*01)=Y(A3)-.05
XXX(2*01)=XX(2*01)=X(2*01)=X(A1)-.015
YYY(2*01)=YY(2*01)=Y(2*01)=Y(A3)+.02
XXX(3*01)=XX(3*01)=X(3*01)=X(A1)
YYY(3*01)=YY(3*01)=Y(3*01)=Y(A3)+.05
XXX(4*01)=XX(4*01)=X(4*01)=X(A1)+.015
YYY(4*01)=YY(4*01)=Y(4*01)=Y(A3)+.02
XXX(5*01)=XX(5*01)=X(5*01)=X(A1)-.02
YYY(5*01)=YY(5*01)=Y(5*01)=Y(A3)-.05
XXX(1*02)=XX(1*02)=X(1*02)=X(A2)+.02
YYY(1*02)=YY(1*02)=Y(1*02)=Y(A4)-.05
XXX(2*02)=XX(2*02)=X(2*02)=X(A2)-.015
YYY(2*02)=YY(2*02)=Y(2*02)=Y(A4)+.02
XXX(3*02)=XX(3*02)=X(3*02)=X(A2)
YYY(3*02)=YY(3*02)=Y(3*02)=Y(A4)+.05
XXX(4*02)=XX(4*02)=X(4*02)=X(A2)+.015
YYY(4*02)=YY(4*02)=Y(4*02)=Y(A4)+.02
XXX(5*02)=XX(5*02)=X(5*02)=X(A2)-.02
YYY(5*02)=YY(5*02)=Y(5*02)=Y(A4)-.02
XXX(1*03)=XX(1*03)=X(1*03)=X(A1)+.02
YYY(1*03)=YY(1*03)=Y(1*03)=Y(A3)-.02
XXX(2*03)=XX(2*03)=X(2*03)=X(A1)-.015
YYY(2*03)=YY(2*03)=Y(2*03)=Y(A3)+.02
XXX(3*03)=XX(3*03)=X(3*03)=X(A1)
```



$Y(Y(3 \cdot 03) = Y(A3) + .05$   
 $Y(X(4 \cdot 03) = X(A1) + .015$   
 $Y(X(X(4 \cdot 03) = Y(A3) + .02$   
 $Y(X(X(5 \cdot 03) = X(A1) - .02$   
 $Y(X(X(5 \cdot 03) = Y(A3) - .05$   
 $Y(X(X(1 \cdot 04) = Y(A2) + .02$   
 $Y(X(X(1 \cdot 04) = Y(A4) - .05$   
 $Y(X(X(2 \cdot 04) = X(A2) - .015$   
 $Y(X(X(2 \cdot 04) = Y(A4) + .02$   
 $Y(Y(3 \cdot 04) = Y(A2) + .05$   
 $Y(X(X(3 \cdot 04) = Y(A2) + .015$   
 $Y(X(X(4 \cdot 04) = Y(A4) + .02$   
 $Y(X(X(5 \cdot 04) = Y(A4) - .02$   
 $Y(X(X(1 \cdot 05) = Y(A2) - .05$   
 $Y(X(X(1 \cdot 05) = Y(G1) + .02$   
 $Y(X(X(1 \cdot 05) = Y(G3) - .05$   
 $Y(X(X(2 \cdot 05) = Y(G1) + .015$   
 $Y(X(X(2 \cdot 05) = Y(G3) + .02$   
 $Y(X(X(3 \cdot 05) = Y(G1) - .05$   
 $Y(X(X(3 \cdot 05) = Y(G3) + .015$   
 $Y(X(X(4 \cdot 05) = Y(G1) + .02$   
 $Y(X(X(4 \cdot 05) = Y(G3) - .05$   
 $Y(X(X(5 \cdot 05) = Y(G1) + .02$   
 $Y(X(X(5 \cdot 05) = Y(G3) - .05$   
 $Y(X(X(1 \cdot 06) = Y(G1) + .02$   
 $Y(X(X(1 \cdot 06) = Y(G3) - .05$   
 $Y(X(X(2 \cdot 06) = Y(G1) - .02$   
 $Y(X(X(2 \cdot 06) = Y(G3) + .02$   
 $Y(X(X(3 \cdot 06) = Y(G1) + .02$   
 $Y(X(X(3 \cdot 06) = Y(G3) - .05$   
 $Y(X(X(4 \cdot 06) = Y(G1) + .015$   
 $Y(X(X(4 \cdot 06) = Y(G3) + .02$   
 $Y(X(X(5 \cdot 06) = Y(G1) - .02$   
 $Y(X(X(5 \cdot 06) = Y(G3) + .02$   
 $Y(X(X(1 \cdot 07) = Y(G1) - .05$   
 $Y(X(X(1 \cdot 07) = Y(G3) + .02$   
 $Y(X(X(2 \cdot 07) = Y(G1) + .015$   
 $Y(X(X(2 \cdot 07) = Y(G3) + .02$   
 $Y(X(X(3 \cdot 07) = Y(G1) - .05$   
 $Y(X(X(3 \cdot 07) = Y(G3) + .02$   
 $Y(X(X(4 \cdot 07) = Y(G1) + .05$   
 $Y(X(X(4 \cdot 07) = Y(G3) + .015$   
 $Y(X(X(4 \cdot 07) = Y(G1) + .02$   
 $Y(X(X(4 \cdot 07) = Y(G3) - .05$   
 $Y(X(X(5 \cdot 07) = Y(G1) + .02$   
 $Y(X(X(5 \cdot 07) = Y(G3) - .05$   
 $Y(X(X(1 \cdot 08) = Y(G2) + .02$   
 $Y(X(X(1 \cdot 08) = Y(G4) - .05$   
 $Y(X(X(2 \cdot 08) = X(G2) - .015$















```
YY(4,22)=YY(4,22)=Y(S8)+.02
XX(5,22)=XX(5,22)=X(S2)+.02
YY(5,22)=YY(5,22)=Y(S8)-.05
DO 20 I=1,22
DO 20 J=2,6
K=J-1
IMAN(J,I)=IPACK(XXX(K,I),YYY(K,I),IMD(K))
RETURN
END
20
```



CC SUBROUTINE GINP  
CC WRITTEN BY A. WONG. EE DEPT. INPUT FROM GRAPHICS TERMINAL  
CC PURPOSE: PROVIDE NAMELIST INPUT FROM GRAPHICS TERMINAL  
CC

SUBROUTINE GINP(IDEV,ITDIR,IBLK,IBUF)  
DIMENSION IBUF(1),ITDIR(1),IBLK,IBUF  
IB=IBLK+1  
NULL=-1  
IF(IBUF(1).NE.-1)GO TO 100  
1 ENCODE(16,10,IBUF)  
FORMAT('NAMELIST INPUT')  
CALL TEXTO(IDEV,IBUF,4,18,28,3,3,IER)  
IF(IER.NE.0)OUTPUT(101)IER,GINP1  
RFTURN  
CALL TEXTO(IDEV,NULL,1,18,28,3,3,IER)  
IF(IER.NE.0)OUTPUT(101)IER,NULL1  
CALL TEXTO(IDEV,NULL,1,24,43,3,3,IER)  
IF(IER.NE.0)OUTPUT(101)IER,NULL2  
RFTURN  
CALL TEXTR(IDEV,NULL,1,24,43,3,3,IER)  
IF(IER.NE.0)OUTPUT(101)IER,GINP2  
100 IF(MOD(ITDIR(1B),8).EQ.0)GO TO 110  
CALL TEXTR(IDEV,IBUF,24,0,IB,IER)  
IF(IER.NE.0)OUTPUT(101)IER,GINP3  
RETURN  
END



SUBROUTINE GINPUT  
WRITTEN BY A. WONG, EE DEPT. USNPGS  
PURPOSE: PROVIDE NAMELIST INPUT FROM GRAPHICS TERMINAL

0	\$GINPUT	PZE	BRM	0	9SETUPN	0
3	PZE	PZE	LDP	1	TEXT0 &3	
0	PZE	PZE	STD	1	IBLK	
0	PZE	PZE	LDA	1	NPADR	
0	PZE	PZE	LDA	1	BLCK	
0	PZE	PZE	ADD	1	NPADR	
0	PZE	PZE	STA	1	BUF	
0	PZE	PZE	LDA	1	IBUF	
0	PZE	PZE	BRM	1	R>NES	
0	PZE	PZE	BRM	1	IBUF	
0	PZE	PZE	BRM	1	TEXT0	
0	PZE	PZE	LDA	1	NPADR	
0	PZE	PZE	ADD	1	READ	
0	PZE	PZE	STA	1	PATCH	
0	PZE	PZE	LDA	1	BRM	
0	PZE	PZE	XMA	1	PATCH	
0	PZE	PZE	STA	1	BRM	
0	PZE	PZE	LDA	1	#101	
0	PZE	PZE	STA	1	IBUF	
0	PZE	PZE	BRM	1	9INPUT	
0	PZE	PZE	STZ	1	IBUF	
0	PZE	PZE	BRM	1	TEXT0	
0	PZE	PZE	LDA	1	BRM	
0	PZE	PZE	XMA	1	PATCH	
0	PZE	PZE	STA	1	BRM	
0	PZE	PZE	BRR	1	9INPUT	
0	GINP	BRM	BRM	1	TEXT0	0







## INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3. Chief of Naval Personnel Pers 11b Department of the Navy Washington, D. C. 20370	1
4. Naval Postgraduate School Department of Operations Research and Administrative Sciences Monterey, California 93940	1
5. Professor R. W. Butterworth, Code 55 Bd Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
6. Professor W. M. Raike, Code 55 Rj Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
7. Professor P. A. W. Lewis, Code 55 Lw Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
8. Professor D. P. Gaver, Code 55 Gv Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
9. Professor A. F. Andrus, Code 55 As Department of Operations Research and Administrative Sciences Naval Postgraduate School Monterey, California 93940	1



10. Lieutenant Cyrus Michael Riddell  
1615 92 North East  
Bellevue, Washington 98004

1



## UNCLASSIFIED

Security Classification

## DOCUMENT CONTROL DATA - R &amp; D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)		2a. REPORT SECURITY CLASSIFICATION	
Naval Postgraduate School Monterey, California 93940		Unclassified	
3. REPORT TITLE			
A Queuing Model Simulation with Dynamic Graphical Display			
4. DESCRIPTIVE NOTES (Type of report and, inclusive dates) Master's Thesis; March 1973			
5. AUTHOR(S) (First name, middle initial, last name) Cyrus Michael Riddell			
6. REPORT DATE March 1973		7a. TOTAL NO. OF PAGES 90	7b. NO. OF REFS
8a. CONTRACT OR GRANT NO.		9a. ORIGINATOR'S REPORT NUMBER(S)	
b. PROJECT NO.			
c.		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	
d.			
10. DISTRIBUTION STATEMENT Approved for public release; distribution unlimited.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940	
13. ABSTRACT  The purpose of this thesis is to provide the student of queuing systems with a vehicle through which a better understanding of the interrelationships between stochastic processes and queuing systems can be achieved.			



## UNCLASSIFIED

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Queues						
Simulation						
Computer Graphics						



2 JUN 76

24425

Thesis

143254

R425 Riddell

c.1. A queuing model simulation with dynamic graphical display.

2 JUN 76

24425

Thesis

143254

R425 Riddell

c.1. A queuing model simulation with dynamic graphical display.

thesR425  
A queueing model simulation with dynamic



3 2768 000 99057 6  
DUDLEY KNOX LIBRARY